ON THE NONNEGATIVE LEAST SQUARES ALGORITHM

A Thesis
Presented to
The Academic Faculty

by

Cláudio Prata Santiago

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology
December 2009

ON THE NONNEGATIVE LEAST SQUARES ALGORITHM

Approved by:

Earl Barnes, Committee Chair
School of Industrial and Systems Engineering
Georgia Institute of Technology

Faiz Al-Khayyal
School of Industrial and Systems Engineering
Georgia Institute of Technology

Arkadi Nemirovski
School of Industrial and Systems Engineering
Georgia Institute of Technology

Joel Sokol
School of Industrial and Systems Engineering
Georgia Institute of Technology

Guillermo H. Goldsztein
School of Mathematics
Georgia Institute of Technology

Date Approved: 7 August, 2009

To my beloved parents,

Sílvio and Cláudia.

ACKNOWLEDGEMENTS

I thank the Almighty Heavenly Father YHWH for the strength that He gave me throughout my life.

I thank my parents, Sílvio and Cláudia for having devoted their lives entirely to me and to my brothers.

I cannot find words to express my gratitude to my aunts Eliana and Fernanda for all the support they gave my brothers and I along our entire lives.

I thank my brothers André and Cristiano for their help during my hardest times.

I greatly thank my godparents, Roberto and Lúcia, my uncle Renato and my aunt Maura for all the help they gave my parents, my brothers and I throughout our lives.

I am truly thankful to my advisor, Dr. Earl Barnes, for his patience and dedication to this work.

I am equally indebted to Dan Faissol and Dorothy, Eser Kırkızlar and Helder Inácio. I thank them for their crucial support in all ways during my whole studies at Georgia Tech. I can honestly say I would not have made it this far, were I not able to count on their precious friendship.

I would also not be here without the greatly appreciated support of my master's thesis advisor, Dr. Nelson Maculan, and my undergraduate advisor, Dr. Marcos Negreiros. I also wish to extend my gratitude to Dr. Abilio Lucena and Dr. Maria Helena for supporting my application to the Ph.D. program at Georgia Tech.

I greatly thank the members of my Thesis committee: Dr. Faiz Al-Khayyal, Dr. Arkadi Nemirovski, Dr. Joel Sokol and Dr. Guillermo Goldsztein, for being extremely understanding and for the valuable suggestions during both my proposal and my defense.

I am also indebted to Dr. Shabbir Ahmed and Dr. Alex Shapiro, from ISyE Department, Dr. Hao Min Zhou and Dr. Christopher Heil, from Department of Mathematics, for their restless efforts while teaching their courses.

I thank the friends that dedicated time, even from far away, to send me their words of encouragement, especially Sergio Monteiro, Fátima Marques, Eduardo Joaquim, Tiberius Bonates, Gabriel Paillard, Jorge Bergson and Gerson Castro.

I thank also Dr. Gary Parker for his willingness to help all the students in every possible way. I would like to extend my gratitude to the ISyE staff, especially Pam Morrison and Valarie DuRant-Modeste.

It is with great pleasure that I would like to mention some of my friends that contributed to this great achievement: Abhi, Adaora, Ali, Aly, Alejandro, Amandeep, Bernardo and Daniela, Clarence, Dan Steffy, Deniz, Dylan, Fatma, Juan Pablo and Johana, Melike, Norbert, Örsan, Ricardo and Marta, Vijay, Yuri.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

In this document, we study the use of the nonnegative least squares method to solve the assignment problem (via the primal-dual method), the maximum cardinality matching and also the minimization of separable differentiable convex functions. In addition to that, we propose a new method whereby the solution of the assignment problem can be found by solving a single nonnegative least squares problem.

The motivation for studying the nonnegative least squares primal-dual algorithm comes from its efficiency in solving linear programming problems as was shown empirically in [6]. (See also [19]). The first primal-dual linear programming algorithm was formulated for solving the assignment problem. It was discovered by Harold Kuhn [21], and called the Hungarian method in honor of two Hungarian mathematicians whose work was used in developing the algorithm. The Hungarian method proved to be an effective method for solving assignment problems. Any new primal-dual algorithm must be effective on some class of problems to be of interest. In particular, it must be compared to the Hungarian method for the assignment problem. With this in mind, we have tried to determine how the nonnegative least squares primal-dual algorithm relates to the Hungarian method for the assignment problem. We have established several connections between the two algorithms, and more generally, between the nonnegative least squares algorithm and the weighted matching problem on general graphs. In [6], the authors showed that the nonnegative least squares algorithm is a steepest ascent method for solving the dual of a linear programming problem. This means that this method should require fewer steps, on average, than the Hungarian method. That this is the case was shown empirically in [6], and more

evidence that this is the case will be offered in Chapter 1. Our main result is a procedure for obtaining the solution of an assignment problem by solving a single nonnegative least squares problem.

In Chapter 2, we discuss the theory behind the nonnegative least squares algorithm, the primal-dual algorithm and the nonnegative least squares primal-dual algorithm developed by E. Barnes et al. in [6].

In Chapter 3, we devise a fast procedure to compute the unrestricted least squares solution by exploiting the special structure of the incidence matrix of a bipartite graph. First, we show how computing the unrestricted least squares is used in a modified version of the nonnegative least squares algorithm, and, in this way, we derive a very efficient procedure to calculate the new dual direction of the primal-dual method applied to the assignment problem.

In Chapter 4, we explain how to extract a solution for the matching problem from the nonnegative least squares solution. The main idea is to make a connection between solutions obtained by the nonnegative least squares algorithm and the solutions of the maximum cardinality matching problem for the same graph.

In Chapter 5, we look into some theoretical results concerning the solution of minimization of p-norms and separable differentiable convex functions subject to constraint matrices that are incidence matrices.

In Chapter 6, we show that the assignment problem can be reduced to a single nonnegative least squares problem. This means that the primal-dual approach can be made to converge in one step for the assignment problem. This method does not reduce the primal-dual approach to one step for general linear programming problems, but it appears to give a good starting dual feasible point for the general problem.

Throughout this document, we use the acronym NNLS to stand for Nonnegative Least Squares.

CHAPTER I

INTRODUCTION

One of the first linear programming problems studied by the pioneers of this subject is the assignment problem. This problem exhibits massive degeneracy. Every basic feasible solution is degenerate. This causes the Simplex algorithm to perform poorly on these problems. Tests on randomly generated assignment problems show that approximately 90% of Simplex steps are degenerate. That is, 90% of the Simplex steps do not improve the objective function.

In 1956, Kuhn [21] proposed an efficient method for solving the assignment problem. He called it the Hungarian method in honor of two Hungarian mathematicians whose worked influenced his approach. Later, Dantzig and Fulkerson in [11] observed that the Hungarian method could be generalized to any linear programming problem. They called this generalization the primal-dual method. This method is most effective on highly degenerate problems for which a non-simplex method can be used to solve the subproblems that occur in the primal step of the algorithm.

In section 1.1, it will be presented the assignment problem, as it was the original motivation for this work, as well as a brief description of the main methods used to solve it.

In section 1.2, we look into the Hungarian method (see [21]), since it influenced several of the existing methods in Combinatorial Optimization that deal with bipartite graphs. We discuss intuitive ways of modifying it in order to achieve better practical results. This leads to the NNLS approach.

1.1 The Assignment Problem

Consider a set of n workers and n tasks. Suppose that there is a cost c_{ij} to assign worker i to task j. Suppose further that one worker must be assigned to one task and each task must have at least one worker assigned to it. The assignment problem consists of assigning these n workers to n tasks while minimizing the total cost. This problem can be formulated as the following LP problem:

$$
\begin{aligned}
(P) : \min \quad & \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\
\text{s.t.} \quad & \\
& \textstyle\sum_{j=1}^{n} x_{ij} = 1 \ , \ \forall i \\
& \textstyle\sum_{i=1}^{n} x_{ij} = 1 \ , \ \forall j \\
& x \geq 0
\end{aligned}
$$

Given the nature of this problem, bipartite graphs naturally arise when modeling it. Incidence matrices of bipartite graphs have nice combinatorial properties such as unimodularity (see [10], [29], [35] and [33]), leading to more efficient algorithms.

The most famous algorithm for the assignment problem is the Hungarian method developed by Kuhn (see [21]), based on the work by the Hungarian mathematicians König and Egerváry (see [13] and [20] - translation). Munkres in [26] showed that the complexity Hungarian method is $O(n^4)$.

The contribution of Kuhn's Hungarian method goes far beyond its efficiency to solve the assignment problem, since it has bore on a number of algorithms in Combinatorial Optimization that draws on the same principles conceived by it, such as in transportation and network flow problems (see [1] and [22]).

Versions of the primal Simplex method were shown to solve the assignment problem in $O(n^3)$ pivots (see [17], [27]). Akgül in [2] devises a method that requires $O(n^2)$ pivots.

The signature method (a version of the dual Simplex method) developed by Balinski (see [3]) is shown in [14] to take $O(n^3)$ time. More on dual Simplex algorithms can be found in [32], [4], [30], [32] and [24].

The Modified Hung-Rom Algorithm based upon Hirsch-Paths (proposed in [18]) was shown to be very efficient in computational experiments performed in [28]. Infeasible paths were considered to solve the assignment problem in [31].

The auction method devised by Bertsekas (see [7] and [8]) starts from an incomplete assignment (i.e. there are unassignment nodes) and tries at each iteration to assign an unassignment nodes by a bidding mechanism, that is basically the amount by which the dual solution will increase, while keeping the ϵ-complementary slackness condition (for details see [9]).

All of the aforementioned methods perform primal (or dual) updates that rely on measures of linear decrease (or increase) in the objective function, i.e., the direction is considered by taking into account some type of norm 1 evaluation of increase, decrease or optimality. Therefore, one natural question arises: would it be possible to consider other norms? Would there be an advantage to using other norms?

The primal-dual method provides the perfect framework for considering other measures of increase or decrease, since any strictly convex function that is zero at zero could be used to check feasibility of linear systems. That is exactly the modification that E. Barnes et al. [6] proposed in the primal-dual method in order to obtain a better update direction by using the NNLS algorithm. We expand on this method for the case of the assignment problem by elaborating on the comparison of norm 1 and norm 2 problems when the constraint matrix is an incidence matrix of a graph.

1.2 Motivation

We will motivate the use of a direction computed by the norm 2 by showing a small example that illustrates how superior it can be to the norm 1 update used by the

Hungarian method.

Consider the example on figure 1(a) where there are four workers (denoted by W's) and four jobs (denoted by J's).

In order to give an intuitive idea of the Hungarian method, observe that if any column or any row is decreased (or increased) by a certain amount, any complete assignment will be decreased (increased) by the same amount. For that reason, subtracting or adding the same quantity in any row or column will not change the optimum. In figure 1(b), we can see two different complete assignments and check that the cost of each will vary by the same amount when any number is subtracted or added to all the elements of a row or column.

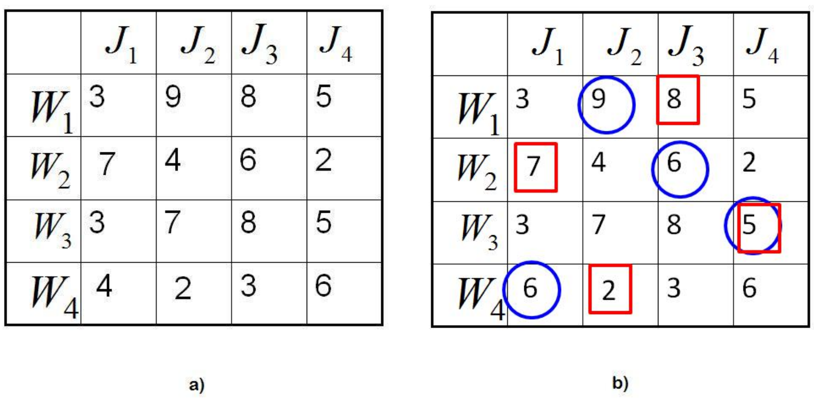

Figure 1: Example of two feasible assignments.

The Hungarian method begins by subtracting the smallest number in each row of the cost matrix from that row. This produces one zero in each row. Now, the smallest number in each column is subtracted from the column. The resulting cost matrix now has at least one zero in each row and column (see figure 2(a)).

After generating one zero in each column and row, the Hungarian method observes that if an assignment can be chosen using only zero costs in the reduced matrix, then this assignment must be optimal. In [13] and [20], König and Egerváry show that the number of lines needed to cover all the zeros is equal to the maximum maximum cardinality matching problem (see figure $2(b)$).

If an assignment cannot be chosen using only the zero costs in the reduced matrix, then we need to generate more zeros by subtracting a certain amount t from every row or column not covered by a line, and adding the same amount to rows and columns covered by lines. This procedure has the effect of subtracting $2t$ to each uncovered reduced cost, and adding $2t$ to each doubly covered reduced cost. If $\bar{c}_{rs}$ is the smallest uncovered reduced cost, then we take $2t = \bar{c}_{rs}$.

Despite not guaranteeing that the number of zeros will decrease, we will generate zeros in places that are 'more likely' to give us an increase in the number of possible assignments.

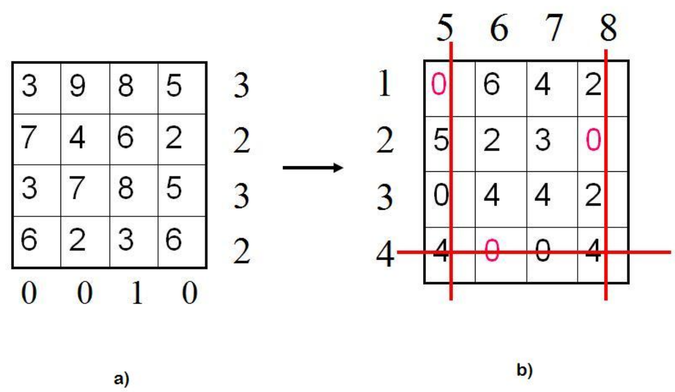

Figure 2: Generating one zero per row and one per column.

Therefore, it is natural to ask: can we improve the performance of the algorithm by subtracting different amounts from row and columns as in figure 5?

In figure and 6, we start with the reduced matrix obtained in figure $2(b)$ and subtract $\frac{t}{3}$ from rows 1 and 3, and from columns 2 and 3. We then subtract $-\frac{t}{3}$ from row 4and column 1. For $t = 6$ we obtain a reduced cost matrix containing a very large number of zeros. In this reduced matrix it is easy to determine an optimal solution.

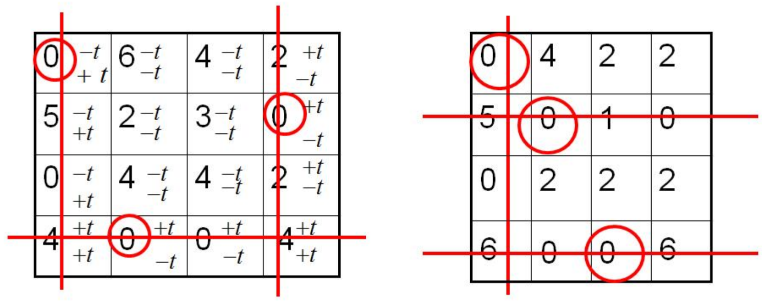

Figure 3: After finding the maximum matching, the dual solution is updated.

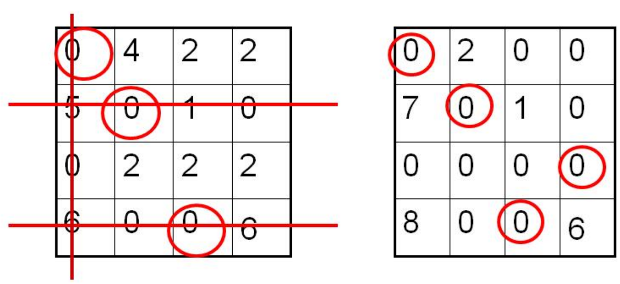

Figure 4: After one more iteration the optimum is reached.

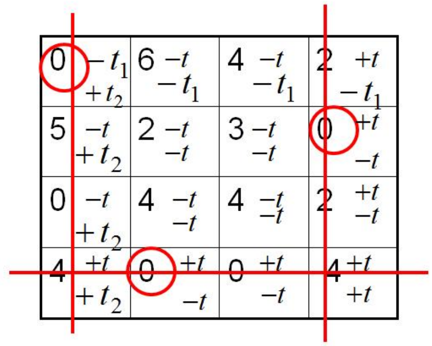

Figure 5: Obtaining a new dual solution.

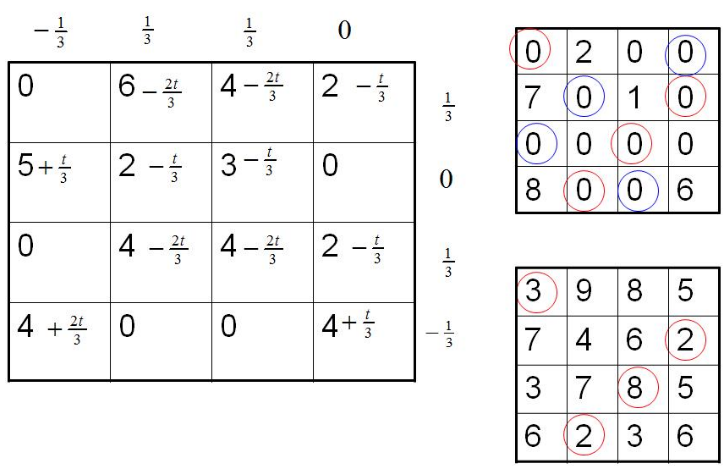

Figure 6: Optimum obtained in the 1st iteration.

CHAPTER II

THE NNLS PRIMAL-DUAL ALGORITHM

In this chapter, we discuss the theory behind the NNLS algorithm (see Leichner et al. [23]), the primal-dual algorithm (developed by Dantzig and Fulkerson in [11]) and the NNLS primal-dual algorithm, proposed by E. Barnes et al. in [6].

In section 2.1, we look into the theory of the NNLS algorithm by first pointing out its similarities with the Simplex method, such as the concept of basis, and then show the proof of its termination.

In section 3.3, we explain in detail the primal-dual algorithm.

In section 2.3, we discuss the NNLS primal-dual algorithm by emphasizing the differences between this special type of primal dual and the traditional one, i.e., the linear primal-dual.

2.1 The NNLS Algorithm

The content of this section is described in more details in [5].

Let A be a $m \times n$ matrix and b be a vector of dimension m.

Consider the following feasibility problem:

$$Ax = b \tag{1}$$

$$x \geq 0 \tag{2}$$

A straightforward way of solving the above problem through linear programming is by solving the following LP problem:

$$(LP) : \min \quad \sum_{j=1}^{n} |s_j|$$

$$\text{s.t.}$$

$$Ax + s = b$$

$$x \geq 0$$

Observe that this norm 1 minimization can be carried out very efficiently by the simplex in most cases. However, there are some constraint matrices for which the Simplex method performs a large number of degenerate pivots, not improving the solution for many iterations, leading to a poor performance.

Our approach to solve the feasibility problem posed by relations 1 and 2 will also be the minimization of a p-norm, but different from the norm 1 considered in (LP), we will consider the norm 2, i.e., we will solve the following problem:

$$(P_{LS}) : \min \quad \frac{1}{2} \sum_{j=1}^{n} s_j^2$$

$$\text{s.t.}$$

$$Ax + s = b$$

$$x \geq 0$$

At first glance, problem (P_{LS}) seems much harder than problem (LP). However, in cases where (LP) is highly degenerate (as pointed out before), it is usually simpler to solve (P_{LS}).

E. Barnes et al. in [6] showed that the normalized direction obtained by (P_{LS}) is the direction of the steepest ascent at π_0 on the dual polyhedron (D). This suggests that the dual direction obtained by (P_{LS}) may be much better in practice than the one obtained by the linear update in (LP). E. Barnes et al. in [6] showed empirically that this is indeed true for some classes of problems.

Since (P_{LS}) is a convex program, KKT conditions are necessary and sufficient for optimality. Thus, the vector (x, s) is a solution for P if and only if there exists π such that:

$$
\begin{aligned}
(D_LS): \quad \pi^t A_j = 0 &\quad \text{if } x_j > 0 \\
\pi^t A_j \leq 0 &\quad \forall j \\
\pi_j = s_j &\quad \forall j
\end{aligned}
$$

The NNLS algorithm starts with a primal feasible solution, i.e., one that is feasible for (P_{LS}), and tries to find a solution for the problem (D_{LS}). The NNLS algorithm is similar to the simplex method in the sense that we have a subset of the columns of A that is a primal feasible basis, and then we move from one primal feasible basis to another. Unlike the simplex method, our 'basis' is not required to be square. The only requirement is that it is composed of linearly independent columns.

Let B be a basis, i.e., a linearly independent subset of the columns of A. Then one crucial step of the nonnegative least squares algorithm is to solve the following problem:

$$\min_x \quad \|Bx - b\|^2$$

Since the columns of B are linearly independent, the solution will be:

$$\overline{x} = B^+ b \quad \text{, where} \quad B^+ = (B^t B)^{-1} B^t$$

The matrix B^+ is called the generalized inverse or pseudo inverse. If B is a basis, we say that it is feasible for (P_{LS}) if we have:

$$\overline{x} = B^+ b > 0 \tag{3}$$

The description of the NNLS algorithm is in algorithm 1.

Theorem 1. *Algorithm 1 terminates with a solution of problem* (P_{LS}).

Algorithm 1 Nonnegative Least Squares Algorithm

1: Let B be a feasible basis for problem P
2: Let I_B be the index set of the columns in B
3: $\overline{x} \leftarrow B^{+}b$
4: $\pi \leftarrow b - B\overline{x}$
5: $S \leftarrow \{j : \pi^t A_j > 0\}$
6: **if** $S = \emptyset$ **then**
7:     Stop: optimal solution found.
8: **end if**
9: Let $k \in S$
10: $d \leftarrow B^{+}A_k$
11: $\underline{\theta} \leftarrow \min_{d_j > 0} \frac{\overline{x}_j}{d_j}$
12: $P \leftarrow I - BB^{+}$
13: $\overline{\theta} \leftarrow \frac{\pi^t A_j}{\|PA_j\|^2}$
14: $\theta \leftarrow \min\{\underline{\theta}, \overline{\theta}\}$
15: **if** $\theta = \overline{\theta}$ **then**
16:     $I_B \leftarrow I_B \cup \{j\}$
17:     **if** $\underline{\theta} = \overline{\theta}$ **then**
18:         $x(\theta) \leftarrow \overline{x}_j - \theta d_j$
19:         $I_B \leftarrow I_B - \{j : x(\theta) = 0\}$
20:     **end if**
21:     $B \leftarrow [A_j], \forall j \in I_B$
22:     Return to 3
23: **else**
24:     $I_B \leftarrow I_B - \{j : \theta = \frac{\overline{x}_j}{d_j}\}$
25:     $B \leftarrow [A_j], \forall j \in I_B$
26:     $\overline{x} \leftarrow B^{+}b$
27:     Return to 10
28: **end if**

Proof. We will show that the algorithm terminates by showing that no basis can be repeated. Since the number of basis is finite, the result follows.

In order to prove that no basis can be repeated, we will show that, if a basis B is updated to $\hat{B}$, then we must have:

$$\min_{x\geq 0} \|\hat{B}x - b\|^2 < \min_{x\geq 0} \|Bx - b\|^2 \; . \tag{4}$$

In order to check 4, we must consider the choice of both stepsizes computed in 11 and 13, according to the comparison in 15.

Let us suppose first that $\theta = \overline{\theta} \leq \underline{\theta}$.

Let B be the current basis and A_j be the entering column.

If $\overline{x}$ is the current primal solution, then

$$\overline{x} = (B^t B)^{-1} B^t b$$

Let $\hat{B}$ be the new basis. Then

$$\min_{x\geq 0} \|\hat{B}x - b\|^2 = \min_{x,t\geq 0} \|Bx + A_j t - b\|^2 \leq \min_{x\geq 0} \|Bx - (b - A_j\theta)\|^2 \; . \tag{5}$$

The solution of the last minimization problem is:

$$\overline{x} = (B^t B)^{-1} B^t (b - A_j\theta) = \overline{x} - \theta d \tag{6}$$

where $d = (B^t B)^{-1} B^t A_j$ is the direction of descent.

Since $\theta \leq \underline{\theta}$, then $x(\theta) \geq 0$. Therefore, $\overline{x}$ is feasible for problem 5. Thus:

$$\begin{aligned} \min_{x\geq 0} \|\hat{B}x - b\|^2 &\leq \|B(\overline{x} - \theta d) - (b - A_j\theta)\|^2 \\ \min_{x\geq 0} \|\hat{B}x - b\|^2 &\leq \|(A_j - Bd)\theta - (b - B\overline{x})\|^2 \end{aligned} \tag{7}$$

Substituting the value of d in 7:

$$\begin{aligned} \min_{x\geq 0} \|\hat{B}x - b\|^2 &\leq \|(A_j - B(B^t B)^{-1} B^t A_j)\theta - (b - B\overline{x})\|^2 \\ \min_{x\geq 0} \|\hat{B}x - b\|^2 &\leq \|(I - B(B^t B)^{-1} B^t) A_j \theta - (b - B\overline{x})\|^2 \end{aligned}$$

Let $P = I - B(B^tB)^{-1}B^t)$. Then:

$$\min_{x \geq 0} \|\hat{B}x - b\|^2 \leq \|PA_j\theta - (b - B\overline{x})\|^2 = \theta^2\|PA_j\|^2 - 2\theta(b - B\overline{x})PA_j + \|b - B\overline{x}\|^2$$

Let $\pi = (b - B\overline{x})^t$. Since $\pi^t P = \pi^t$:

$$\min_{x \geq 0} \|\hat{B}x - b\|^2 \leq \|b - B\overline{x}\|^2 - \frac{(\pi A_j)^2}{\|PA_j\|^2} < \|b - B\overline{x}\|^2 \ .$$

This concludes the proof for the case when $\overline{\theta} \leq \underline{\theta}$.

Now, let us consider the case where $\theta = \underline{\theta} < \overline{\theta}$. In this case, $x(\theta) = \overline{x} - \theta d$ has some zero components.

Consider the following function $h(.)$:

$$h(t) = \min_{x} \|Bx + A_j t - b\|^2 = \min_{x} \|Bx - (b - A_j t)\|^2 \text{ , defined for } 0 \leq t \leq \overline{\theta}$$

Let $x(t)$ be the minimum value of $h(.)$:

$$x(t) = (B^tB)^{-1}B^t(b - A_j t) = \overline{x} - td \tag{8}$$

Substituting 8 in the definition of $h(.)$:

$$h(t) = \|B(B^tB)^{-1}B^t(b - A_j t) - (b - A_j t)\|^2 = \|(I - B(B^tB)^{-1}B^t)A_j t - (b - B\overline{x})\|^2$$

Let $P = I - B(B^tB)^{-1}B^t$ and $\pi = (b - B\overline{x})^t$. Then:

$$h(t) = \|tPA_j - (b - B\overline{x})\|^2 = \|tPA_j\|^2 - 2t\pi^t A_j + \|b - B\overline{x}\|^2 \ .$$

The last equality shows that $h(.)$ is a convex quadratic function of t.

Since $h(t)$ achieves its minimum at $t = \overline{\theta}$, then $h(.)$ must be decreasing on $0 \leq t \leq \overline{\theta}$.

Let $\hat{B}$ be the matrix obtained from B by dropping the columns corresponding to zero components of $x(\theta)$. In particular, we have that:

$$h(t) = \|B\overline{x} - b\|^2 = \min x\|Bx - b\|^2 > h(\theta) = \min x\|\hat{B}x + A_j\theta - b\|^2 \ .$$

Now, let us consider the following function $g(.)$:

$$g(t) = \min_x \|\hat{B}x + A_j t - b\|^2 = \min_x \|\hat{B}x - (b - A_j t)\|^2 \text{ , defined for } t \geq 0 \text{ .} \tag{9}$$

Let us define the following

$$\begin{aligned} \hat{P} &= I - \hat{B}(\hat{B}^t\hat{B})^{-1}\hat{B}^t \\ \hat{x} &= \hat{B}(\hat{B}^t\hat{B})^{-1}\hat{B}^t b \\ \hat{\pi} &= (b - \hat{B}\hat{x})^t \text{ .} \end{aligned}$$

Then:

$$g(t) = \|t\hat{P}A_j - (b - \hat{B}\hat{x})\|^2 = \|t\hat{P}A_j\|^2 - 2t\hat{\pi}^t A_j + \|b - \hat{B}\hat{x}\|^2 \text{ .} \tag{10}$$

Thus, $h(.)$ is also a convex quadratic function of t.

Since the columns of $\hat{B}$ are a subset of the columns of B and $\overline{x} > 0$, we must have:

$$\begin{aligned} g(0) &= \min_x \|\hat{B}x - b\|^2 > \min_x \|Bx - b\|^2 = h(0) \\ g(0) &> h(0) \text{ .} \end{aligned}$$

By construction, we also have $g(\theta) = h(\theta)$. Now, consider the following quadratic function:

$$q(t) = g(t) - h(t) \text{ , defined for } t \geq 0 \text{ .}$$

$$\begin{aligned} g(0) > h(0) &\Rightarrow q(0) > 0 \\ g(\theta) = h(\theta) &\Rightarrow q(\theta) = 0 \end{aligned}$$

If q is a quadratic function, with $q(0) > 0$ and $q(\theta) = 0$, then we must have $q^{'}(\theta) \leq 0$, i.e. $g^{'}(\theta) - h^{'}(\theta) \leq 0$.

Since $h(t)$ achieves its minimum value at $t = \overline{\theta} > \theta = \underline{\theta}$, then $h^{'}(\theta) < 0$. Therefore:

$$g^{'}(\theta) - h^{'}(\theta) \leq 0 \Rightarrow g^{'}(\theta) \leq h^{'}(\theta) < 0 \Rightarrow g^{'}(\theta) < 0$$

Thus, $g(.)$ is decreasing at $\underline{\theta}$. Let $\hat{d} = \hat{B}(\hat{B}^t\hat{B})^{-1}\hat{B}^tA_j$. Then, the solution of 10 is $x^*(t) = \hat{x} - t\hat{d}$ and we have:

$$g(t) = \|\hat{P}A_j\|^2t^2 - 2t(b - B\hat{x})^tA_j + \|b - B\hat{x}\|^2 \ .$$

Its minimum is at the point $t = \overline{\xi}$:

$$\begin{aligned} g^{'}(\overline{\xi}) &= 0 \\ 2\|\hat{P}A_j\|^2\overline{\xi} - 2(b - B\hat{x})^tA_j &= 0 \\ \overline{\xi} = \frac{(b - B\hat{x})^tA_j}{\hat{P}A_j\|^2} &> \underline{\theta} > 0 \end{aligned}$$

Let $\underline{\xi} = \min \frac{\hat{x}_i}{\hat{d}_i}, \forall i$.

Then, if $\overline{\xi} \leq \underline{\xi}$, then we add the column A_j to the basis $\hat{B}$ and set the new basis to be $[\hat{B}, A_j]$ and return to step 3. If we have $\overline{\xi} > \underline{\xi}$, then we need to remove from $\hat{B}$ the columns corresponding to zero components of $\hat{x}(\hat{t})$ and continue to increase t in the function 9. Each time one of the components of $\hat{x}(t)$ is zero, we drop the corresponding column of the basis $\hat{B}$. By repeating this procedure, we will eventually be able to increase the value of t to the point where $g(t)$ achieves its minimum value, since

$$g(t) < h(0) = \|B\overline{x} - b\|^2 < \|b\|^2 \text{ because } \overline{x} \neq 0 \ .$$

$\square$

2.2 *The Primal-Dual Algorithm*

Let A be a $m \times n$ matrix, b be a vector of dimension m and c be a vector of dimension n. We are interested in solving the following problem:

$$\begin{aligned} (P) : \min \quad & \textstyle\sum_{j=1}^n c_jx_j \\ \text{s.t.} \quad & \\ & Ax = b \\ & x \geq 0 \end{aligned}$$

The primal-dual approach to (P) is most interesting in cases where (P) is highly degenerate and the Simplex method performs a large number of pivots without improving the objective. The primal-dual approach produces a sequence of dual feasible vectors that strictly increases the dual objective.

Since we are only interested in computational methods for solving this problem, we restrict our attention to the case where (P) has a solution. We also assume $b \neq 0$.

Before describing the primal-dual algorithm to solve (P), we need the following lemma:

Lemma 2. *$\exists j$ such that $b^t A_j > 0$.*

Proof. Let x be a feasible point for (P).

Then, multiplying the constraint $Ax = b$ on both sides by b we have:

$$b^t Ax = \sum_{j=1}^{n} b^t A_j x_j = b^t b > 0 \quad \rightarrow \quad b^t A_j x_j > 0 \text{ for some } j \tag{11}$$

$$\rightarrow \quad b^t A_j > 0 \text{ for some } j\ . \tag{12}$$

□

Now consider the dual of the problem (P):

$$(D) : \max \qquad \pi^t b$$

$$\text{s.t.}$$

$$\pi^t A \leq c^t$$

The primal-dual algorithm needs a dual feasible vector π_0 to (D). Therefore, in order to apply the primal-dual method, we have to be able to find a dual feasible point easily. We assume that such point π_0 is given.

We need π_0 to be on the boundary of the set $\pi^t A \leq c^t$, since the primal-dual method relies on complementary slackness conditions. If π_0 is an interior point, then we need to need to perform the following update:

$$\begin{aligned} \pi_0 &\leftarrow \pi_0 + \theta b \\ \text{where} \quad \theta &= \min_{\{j|\rho^t A_j > 0\}} \tfrac{c_j - \pi_0^t A_j}{\rho^t A_j} \end{aligned}$$

Let $E = \{A_j | \pi_0^t A_j = c_j\}$ and $I = \{j | A_j \in E\}$. Let x_E be the vector composed of the components of x that correspond to the column in E.

Consider the following feasibility problem:

$$Ex_E = b \tag{13}$$

$$x \geq 0 \tag{14}$$

The previous feasibility problem can be solved by considering the following norm 1 minimization:

$$\begin{aligned} (P_E) \min \quad & \textstyle\sum_{j=1}^{n} |s_j| \\ \text{s.t.} \quad & \\ & Ex_I + Is = b \\ & x_I \geq 0 \end{aligned}$$

If $\exists x_E^*$ feasible for (13-14), then (P_E) has a feasible solution of value zero, then setting $x_j^* = 0$, $\forall j \notin I$, we have a feasible primal-dual pair (π_0, x^*) such that complementary slackness holds: this implies optimality and we are done.

If problem (13-14) is infeasible, then (P_E) has a nonzero solution. Then we need to find an increasing dual direction, that is, we need to find a direction ρ such that for $t > 0$ and sufficiently small we have:

$$\pi = \pi_0 + t\rho \qquad \text{is dual feasible, and} \qquad \pi_1 b > \pi_0 b \ .$$

Farkas' lemma assures us the existence of such a direction, i.e., if problem (13-14) is infeasible, then there must be a vector ρ such that:

$$(D_E) \quad \rho^t E \leq 0$$
$$\rho^t b > 0$$

Lemma 3. *The vector ρ feasible for problem D_E is an strictly increasing dual feasible direction.*

Proof. For any $t > 0$ we have:

$$\rho^t E \leq 0 \quad \Rightarrow \quad t\rho^t E \leq 0$$

If $j \notin I$, then we have $\pi_0 A_j < c_j$. Thus $(\pi_0 + t\rho^t)A_j < c_j$ for $t > 0$ sufficiently small.

$$(\pi_0 + t\rho^t)A_j \leq c_j \ , \ j \in I$$
$$(\pi_0 + t\rho^t)A_j < c_j \ , \ j \notin I$$

Therefore, for $t > 0$ sufficiently small we have:

$$(\pi_0 + t\rho^t)A \leq c^t$$

This proves that the direction ρ will lead to a dual feasible point for some $t > 0$. This direction is strictly increasing, since

$$\rho^t b > 0 \quad \Rightarrow \quad (\pi_0 + t\rho)^t b > \pi_0^t b$$

□

The power of the primal-dual algorithm lies in the fact that the cost of the dual solution is strictly improved at each iteration. However, each iteration is complex since we need to find a vector ρ feasible for problem (D_E). For more details on the primal-dual algorithm, see [29].

2.3 The NNLS Primal-Dual Algorithm

The linear primal-dual algorithm finds the dual increasing direction ρ. The linear primal-dual algorithm finds ρ by solving a linear programming problem:

$$\begin{aligned} \min \quad & \textstyle\sum_{j=1}^{n} |s_j| \\ \text{s.t.} \quad & \\ & Ex_I + Is = b \\ & x_I \geq 0 \end{aligned}$$

The previous norm 1 minimization can be modeled as the standard linear programming problem:

$$\begin{aligned} (LP_E) : \min \quad & \textstyle\sum_{j=1}^{n} s_j^+ + s_j^- \\ \text{s.t.} \quad & \\ & Ex_I + s^+ - s^- = b \\ & x, s^+, s^- \geq 0 \end{aligned}$$

It turns out that if (P) is degenerate, then so is (LP_E), since the column of E are a subset of the columns of A.

In 1992, Dantzig proposed a non-simplex method for solving the general phase I problem, stated in the form

$$\begin{aligned} \min \quad & \textstyle\sum_{j=1}^{n} \frac{1}{2} s_j^2 \\ \text{s.t.} \quad & \\ & Ax_I + s = b \\ & x, \geq 0 \ . \end{aligned}$$

This method (the NNLS) converges fast, even for degenerate problems.

Motivated by this, Barnes et al. [6] modified the traditional primal-dual algorithm by replacing problem (P_E) with the NNLS problem:

$$(LS_E) : \min \quad \sum_{j=1}^{n} \frac{1}{2} s_j^2$$

$$\text{s.t.}$$

$$Ex_I + s = b$$

$$x, \geq 0$$

They showed that this version of the primal-dual algorithm amounts to applying steepest ascent to maximizing the dual of (P), and that it is therefore in general faster than the traditional primal-dual approach.

CHAPTER III

THE NNLS ALGORITHM ON BIPARTITE GRAPHS

In this chapter, we explain the NNLS algorithm and propose a slight modification of the NNLS algorithm that does not require the computation of projection matrices that appear in the original description of the algorithm (see chapter 2). In addition to that, we devise a different way of solving the linear systems that arise at every iteration of the NNLS algorithm. Broadly speaking, we make use of the special structure of the incidence matrix of bipartite graphs in order to obtain the solution of the linear systems in linear time. In addition to that, we show how these results can be used efficiently to solve the assignment problem.

In section 3.1, we present the NNLS along with some minor modifications that eliminate the computation of the orthogonal projection matrix.

In section 3.2, we discuss in detail the crucial alterations to be made in order to improve the performance of the algorithm, by showing how the systems of linear equations that arise in the NNLS algorithm can be solved in linear time for the assignment problem.

In section 3.3, we discuss in detail how the results obtained in sections 3.1 and 3.2 are to be used in the framework of the NNLS primal-dual (discussed in chapter 2) in order to solve the assignment problem by means of an example.

3.1 Avoiding the Computation of the Orthogonal Projection Matrix

The most costly steps in algorithm 1 are the solution of the system of linear equations through the generalized inverse and the computation of the orthogonal projection matrix P. However, we can modify algorithm 1 such that we no longer need to compute the projection matrix P. This is achieved by simply observing that the matrix P is computed only in order to check whether adding the current column selected on 5 (that violates dual feasibility) will make the current basis primal infeasible. If so, some adjustments must be made, i.e., some of the columns on B will be dropped and a new improving direction will be computed in 10.

Let $\overline{x}$ be as in (3). Suppose that $\pi^t A_k > 0$, i.e., column A_k will enter the basis. Let $d = B^+ A_k$. Then we have:

Theorem 4. *Let B be the current basis. Suppose that B has p columns and Let $\overline{x}$ denote the solution of $\min_x \|Bx - b\|^2$ and $x(\theta) = \overline{x} - \theta d$.*
If $\overline{\theta} = \frac{\pi^t A_k}{\|PA_k\|^2}$, then $x(\overline{\theta})$ contains the first $p-1$ coordinates of the solution of

$$\min_x \|\overline{B}x - b\|^2 \qquad , \textit{ where} \qquad \overline{B} = [B, A_k]$$

Proof. Let $E = B^t B$ and $\overline{E} = (\overline{B}^t \overline{B})$. Therefore:

$$\overline{x} = \overline{E}^{-1} \overline{B}^t b$$

Factorizing the matrix $\overline{E}$ we have:

$$\overline{E} = \begin{pmatrix} B^t \\ A_k^t \end{pmatrix} \begin{pmatrix} B & A_k \end{pmatrix} = \begin{pmatrix} E & v \\ v^t & A_k^t A_k \end{pmatrix}$$

where $v = B^t A_k$.

Factorizing the matrix $\overline{E}$ we have:

$$\begin{pmatrix} E & v \\ v^t & A_k^t A_k \end{pmatrix} = \begin{pmatrix} I_p & 0 \\ v_t E^{-1} & 1 \end{pmatrix} \begin{pmatrix} E & 0 \\ 0 & \chi_E \end{pmatrix} \begin{pmatrix} I_p & E^{-1}v \\ 0 & 1 \end{pmatrix}$$

where χ_E be the Shur complement of E:

$$\chi_E = A_k^t A_k - v^t E^{-1} v$$

Computing the inverse of $\overline{E}$ we have:

$$\begin{pmatrix} E & v \\ v^t & A_k^t A_k \end{pmatrix}^{-1} = \left(\begin{pmatrix} I_p & 0 \\ v_t E^{-1} & 1 \end{pmatrix} \begin{pmatrix} E & 0 \\ 0 & \chi_E \end{pmatrix} \begin{pmatrix} I_p & E^{-1}v \\ 0 & 1 \end{pmatrix} \right)^{-1}$$

$$\begin{pmatrix} E & v \\ v^t & A_k^t A_k \end{pmatrix}^{-1} = \begin{pmatrix} I_p & E^{-1}v \\ 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} E & 0 \\ 0 & \chi_E \end{pmatrix}^{-1} \begin{pmatrix} I_p & 0 \\ v_t E^{-1} & 1 \end{pmatrix}^{-1}$$

$$\begin{pmatrix} E & v \\ v^t & A_k^t A_k \end{pmatrix}^{-1} = \begin{pmatrix} I_p & -E^{-1}v \\ 0 & 1 \end{pmatrix} \begin{pmatrix} E^{-1} & 0 \\ 0 & \chi_E^{-1} \end{pmatrix} \begin{pmatrix} I_p & 0 \\ -v_t E^{-1} & 1 \end{pmatrix}$$

$$\begin{pmatrix} E & v \\ v^t & A_k^t A_k \end{pmatrix}^{-1} = \begin{pmatrix} E^{-1} + E^{-1}v\chi_E^{-1}v^t E^{-1} & -E^{-1}v\chi_E^{-1} \\ -\chi_E^{-1}v^t E^{-1} & \chi_E^{-1} \end{pmatrix}$$

We know that the solution of $\min_x \|\overline{B}x - b\|^2$ is:

$$\hat{x} = \overline{E}^{-1}\overline{B}^t b$$

Therefore, using the previous derivation:

$$\hat{x} = \overline{E}^{-1}\overline{B}^t b = \begin{pmatrix} E & v \\ v^t & A_k^t A_k \end{pmatrix}^{-1} \begin{pmatrix} B^t b \\ A_k^t b \end{pmatrix}$$

$$= \begin{pmatrix} E^{-1} + E^{-1}v\chi_E^{-1}v^t E^{-1} & -E^{-1}v\chi_E^{-1} \\ -\chi_E^{-1}v^t E^{-1} & \chi_E^{-1} \end{pmatrix} \begin{pmatrix} B^t b \\ A_k^t b \end{pmatrix}$$

Now, let

$$\hat{x}_\theta = \begin{pmatrix} \hat{x}_1 \\ . \\ . \\ . \\ \hat{x}_{p-1} \end{pmatrix}$$

$$
\begin{aligned}
\hat{x}_\theta &= (E^{-1} + E^{-1}v\chi_E^{-1}v^tE^{-1})B^tb - E^{-1}v\chi_E^{-1}A_k^tb \\
\hat{x}_\theta &= E^{-1}B^tb + E^{-1}v\chi_E^{-1}v^tE^{-1}B^tb - E^{-1}v\chi_E^{-1}A_k^tb \\
\hat{x}_\theta &= \overline{x} + (E^{-1}vv^tE^{-1}B^tb - E^{-1}vA_k^tb)\chi_E^{-1} \\
\hat{x}_\theta &= \overline{x} + \frac{E^{-1}vv^tE^{-1}B^tb - E^{-1}vA_k^tb}{\chi_E}
\end{aligned}
$$

Since $\chi_E = \|PA_k\|^2$:

$$
\hat{x}_\theta = \overline{x} + \frac{E^{-1}vv^tE^{-1}B^tb - E^{-1}vA_k^tb}{\|PA_k\|^2} \tag{15}
$$

The current direction of increase is given by:

$$
d = E^{-1}B^tA_k = E^{-1}v \tag{16}
$$

Substituting (27) in (26):

$$
\begin{aligned}
\hat{x}_\theta &= \overline{x} + \frac{dv^tE^{-1}B^tb - dA_k^tb}{\|PA_k\|^2} \\
\hat{x}_\theta &= \overline{x} + \frac{v^tE^{-1}B^tb - A_k^tb}{\|PA_k\|^2}d \\
\hat{x}_\theta &= \overline{x} - \frac{A_k^t\pi}{\|PA_k\|^2}d \\
\hat{x}_\theta &= \overline{x} - \overline{\theta}d \\
\hat{x}_\theta &= x(\overline{\theta})
\end{aligned}
$$

□

Corollary 5. *Let $\hat{x}_\theta$ be the first $p-1$ coordinates of the solution of*

$$
\min_x \|\overline{B}x - b\|^2 \qquad , \textit{ where } \qquad \overline{B} = [B, A_k]
$$

then we have:

$$
\hat{x}_\theta \geq 0 \quad \Longleftrightarrow \quad \overline{\theta} \leq \underline{\theta}
$$

Proof. From theorem (4):

$$\hat{x}_\theta \geq 0 \quad \Longleftrightarrow \quad x(\overline{\theta}) \geq 0 \quad \Longleftrightarrow \quad \overline{\theta} \leq \underline{\theta}$$

□

Corollary (5) tells us that the feasibility of the first $p-1$ coordinates of the new basis is equivalent to comparing $\overline{\theta}$ and $\underline{\theta}$. That means that if we compute the new solution then any of the $p-1$ coordinates will be less than zero if and only if $\overline{\theta} > \underline{\theta}$.

It is important to notice that corollary (5) does not address the issue of the sign of the last coordinate, i.e., the coordinate of the solution corresponding to the entering column. However, the next lemma will state that this last coordinate will always be positive, once it corresponds to a column that wants to enter the basis.

Using the definition of $\hat{x}$ presented in the proof of theorem (4), since $\hat{x}$ is a $p-$dimensional vector, let $\hat{x}_p$ be its *pth* coordinate. We have the following lemma:

Lemma 6. *Let A_k denote the column that is entering the basis and let $\hat{x}_p$ denote the corresponding component of $\hat{x}$. Then we must have that $\hat{x}_p > 0$.*

Proof.

$$\begin{aligned}
\hat{x}_p &= -\chi_E^{-1} v^t E^{-1} B^t b + \chi_E^{-1} A_k^t b \\
\hat{x}_p &= \chi_E^{-1} (A_k^t b - v^t E^{-1} B^t b) \\
\hat{x}_p &= \chi_E^{-1} A_k^t (b - v^t E^{-1} B^t b) \\
\hat{x}_p &= \chi_E^{-1} A_k^t \pi
\end{aligned}$$

Therefore:

$$\hat{x}_p = \chi_E^{-1} \pi^t A_k > 0$$

The last inequality follows from the fact that $\pi^t A_k > 0$ (since it is an entering column) and $\chi_E > 0$ (lemma on Shur complement). □

Therefore, we do not need to compute both stepsizes, just $\underline{\theta}$. It suffices to compute the new solution (i.e., the least squares solution of the augmented matrix) and check if any of the coordinates is negative ($\overline{\theta} > \underline{\theta}$) or zero ($\overline{\theta} = \underline{\theta}$). The computation of $\underline{\theta}$ is still necessary in case that $\overline{\theta} > \underline{\theta}$, i.e., one of the coordinates is negative, since we need to extract from the current basis the columns for which this stepsize is achieved.

Now, we may present the modified NNLS algorithm as follows.

Algorithm 2 Modified Nonnegative Least Squares Algorithm

Let B be a feasible basis for problem P
Let I_B be the index set of the columns in B
$\overline{x} \leftarrow B^+ b$
$\pi \leftarrow b - B\overline{x}$
$S \leftarrow \{j : \pi^t A_j > 0\}$
if $S = \emptyset$ **then**
    Stop: optimal solution found.
end if
Let $k \in S$
$d \leftarrow B^+ A_k$
$\theta \leftarrow \min_{d_j > 0} \frac{\overline{x}_j}{d_j}$
$\hat{x}_\theta \leftarrow [B, A_k]^+ b$
if $\hat{x}_\theta \geq 0$ **then**
    $I_B \leftarrow I_B \cup \{j\}$
    $I_B \leftarrow I_B - \{j : \hat{x}_\theta = 0\}$
    $B \leftarrow [A_j], \forall j \in I_B$
    Return to 3
else
    $I_B \leftarrow I_B - \{j : \theta = \frac{\overline{x}_j}{d_j}\}$
    $B \leftarrow [A_j], \forall j \in I_B$
    $\overline{x} \leftarrow B^+ b$
    Return to 10
end if

The most costly step in algorithm 2 is the solution of the linear system through the generalized inverse. That means that if we can devise an efficient method to solve $\min_x \|Bx - b\|$, where B is a primal basis for the NNLS, then algorithm 2 will run much faster.

3.2 Solving the Systems of Linear Equations in Linear Time

Let G be the graph on which we want to solve the assignment problem. Let A be the incidence matrix of G. Consider again the following problem:

$$\begin{aligned} (P):\min \quad & \tfrac{1}{2}\textstyle\sum_{j=1}^{n} s_j^2 \\ \text{s.t.} \quad & \\ & Ax + s = b \\ & x \geq 0 \end{aligned}$$

We know from the KKT conditions that the vector (x, s) is a solution for P if and only if there exists π such that:

$$\begin{aligned} (D): \quad & \pi^t A_j = 0 \quad \text{if } x_j > 0 \\ & \pi^t A_j \leq 0 \quad \forall j \\ & \pi_j = s_j \quad \forall j \end{aligned}$$

We will start this section proving a very useful lemma that allows us to calculate the dual solution of the NNLS problem.

Consider the relaxed version of problem P where we drop the nonnegativity constraints:

$$\begin{aligned} (P_R):\min \quad & \tfrac{1}{2}\textstyle\sum_{j=1}^{n} s_j^2 \\ \text{s.t.} \quad & \\ & Ax + s = b \end{aligned}$$

We know from the KKT conditions of this relaxed problem that the vector (x, s) is a solution for P_R if and only if there exists π such that:

$$(D_R): \quad \pi^t A_j = 0 \quad \forall j \tag{17}$$

$$\pi_j = s_j \quad \forall j \tag{18}$$

Let Δ and Ξ be the partitions of G. Observe that Δ and Ξ are well defined since G is bipartite. Let

$$\begin{aligned} \Delta^* &= \textstyle\sum_{j\in\Delta} b_j \\ \Xi^* &= \textstyle\sum_{j\in\Xi} b_j \end{aligned}$$

Then we have the following lemma:

Lemma 7. *Suppose that G is connected and let (x^*, s^*) be the solution of P_R. Then*

$$\begin{aligned} s_j^* &= \tfrac{\Delta^*-\Xi^*}{n} \quad \forall j \in \Delta \\ s_j^* &= \tfrac{\Xi^*-\Delta^*}{n} \quad \forall j \in \Xi \end{aligned}$$

Proof. It is clear from the constraints (17) of problem (D_R) that if (x^*, s^*) be the solution of P_R, with $s_j^* = \gamma$, for $j \in \Delta$, then $s_j^* = -\gamma$, for $j \in \Xi$. Thus, we reduce problem (P_R) to:

$$\begin{aligned} (\overline{P}_R) : \min \quad & \textstyle\sum_{j=1}^{n} s_j^2 \\ \text{s.t.} \quad & \\ & \textstyle\sum_{i\in\delta_k} x_i + \gamma = b_k \quad k \in \Delta \\ & \textstyle\sum_{i\in\delta_k} x_i - \gamma = b_k \quad k \in \Xi \end{aligned}$$

where δ_k is the set of all edges incident with vertex k. Consider the following system of equations:

$$\sum_{i\in\delta_k} x_i + \gamma = b_k \quad k \in \Delta \tag{19}$$

$$\sum_{i\in\delta_k} x_i - \gamma = b_k \quad k \in \Xi \tag{20}$$

Multiplying the equalities in (20) by -1 and summing with the ones in (19), we will obtain:

$$\begin{aligned} n\gamma &= \Delta^* - \Xi^* \\ \gamma &= \tfrac{\Delta^* - \Xi^*}{n} \end{aligned}$$

proving the lemma. □

Corollary 8. *Suppose the G has k connected components and let (Δ_i, Ξ_i), $i = 1, \ldots, k$ be the partitions of these connected connected components. Let (x^*, s^*) be the solution of P_R, where*

$$s^* = (s^1, \ldots, s^k) \text{ where } s^j \in I\!R^{|\Delta_i| + |\Xi_i|},\ i = 1, \ldots, k.$$

Then

$$\begin{aligned} s^i_j &= \tfrac{\Delta^*_i - \Xi^*_i}{|\Delta_i| + |\Xi_i|} \quad \forall j \in \Delta_i,\ i = 1, \ldots, k \\ s^i_j &= \tfrac{\Xi^*_i - \Delta^*_i}{|\Delta_i| + |\Xi_i|} \quad \forall j \in \Xi_i,\ i = 1, \ldots, k \end{aligned}$$

Proof. Apply lemma 7 to each connected component of G. □

The following lemma is applied to the (modified) NNLS algorithm of the previous section.

Lemma 9. *Suppose that the initial basis used in the NNLS algorithm is a forest, then at any iteration, the basis B will NOT contain the edge set of a cycle.*

Proof. Any column that is added to the basis must be a column that is violating dual feasibility (see step 5 of algorithm 1 and step 5 of algorithm 2). It suffices to show that any arc whose corresponding column is chosen does not form a cycle, but this is straightforward, since if we choose an arc with both ends on the same connected component, one end will have value π and the other $-\pi$ (since the graph is bipartite): this implies that the reduced cost of this column (arc) will be zero, contradicting its dual infeasibility. □

Now we are ready to present the main result of this section:

Lemma 10. *Let T be a tree and A its incidence matrix. Then the unrestricted least squares problem $\min_x \|Ax - b\|^2$ can be solved in linear time on the size of A for any b.*

Proof. Let $\overline{x}$ be the solution of $\min_x \|Ax - b\|^2$ and $\pi = b - A\overline{x}$.
We know from lemma 7 that the value of π can be computed in linear time. We will prove that we can also compute the value of x in linear time.
We will prove by induction on the number of edges of T. Suppose T has just one edge, i.e., A is composed of just one column and its vertices are 1 and 2. Suppose, without loss of generality that $\Delta = \{1\}$ and $\Xi = \{2\}$. From lemma 7, we have

$$\Delta^* = b_1 \ , \ \Xi^* = b_2 \qquad \Rightarrow \qquad \pi_1 = \frac{b_1 - b_2}{1} = -\pi_2$$
$$x + \pi_1 = 1 \qquad \Rightarrow \qquad x = 1 - \pi_1 \qquad \Rightarrow \qquad x = 1 - b_1 + b_2$$

Now, let $|T| = n + 1$, i.e., a tree on n edges. Since T is a tree, the exists an edge incident with a leaf. Without loss of generality, let the vertex 1 denote this leaf and A_1 be the column corresponding to the edge that is incident with this vertex. We can compute in exactly $n+1$ steps (just counting the vertices) of the value of Δ^* and Ξ^*. Since we have Δ^* and Ξ^*, using again lemma 7 we have the value of π. Now, since A_1 is a leaf, we have that $x_1 = b_1 - \pi_1$. We can now extract this leaf from T and repeat the same procedure on the subgraph $T - \{A_1\}$. Note that we must not recompute the values of π for the subgraph $T - \{A_1\}$. Observe also that the total number of steps of this algorithm is $n + 1$ to compute Δ^* and Ξ^* and n to compute x values, where n is the number columns of A. Thus, the complexity of the algorithm is $O(n)$. □

Corollary 11. *Let T be a forest and A its incidence matrix. Then the unrestricted least squares problem $\min_x \|Ax - b\|^2$ can be solved in linear time on the size of A for any b.*

Proof. Apply lemma 10 to each connected component of the forest generated by A. □

3.3 Solving the Assignment Problem using the NNLS Primal-Dual Algorithm

We will give an example to show how the theory worked so far could be put together to solve the assignment problem. Recall the example on section 1.2. Let us first label the rows from 1 to 4 and the columns from 5 to 8. We start off in the same way as in the traditional Hungarian method obtaining a dual feasible solution by generating zeros in each row and column (see figure 7).

After generating the zeros, we build the graph on figure 8 and solve the NNLS in order to obtain the dual direction. Figure 9 shows the NNLS solution. After obtaining the direction, we have the 'weights' that we will use to find the amount to subtract or add from each row or column, as illustrated on figure 10.

After the dual update, we will generate a new set of zeros. This means that are graph will have a different set of edges, as shown on figure 11.

Solving the NNLS problem on the new graph will lead us to a complete assignment, since the NNLS solution is zero. Figures 12 and 13 illustrate two possible basis for the NNLS, giving as a result two minimum assignments.

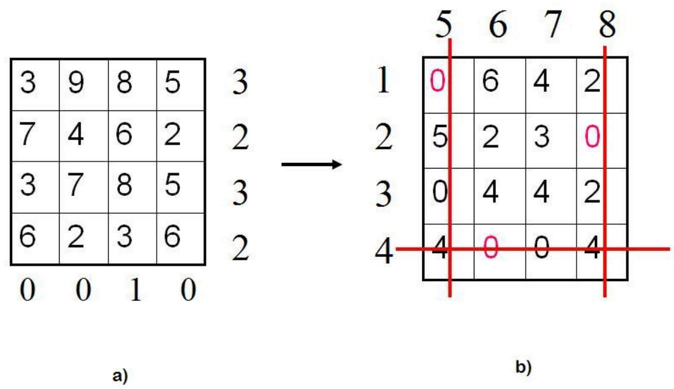

Figure 7: Generating one zero per row and one per column.

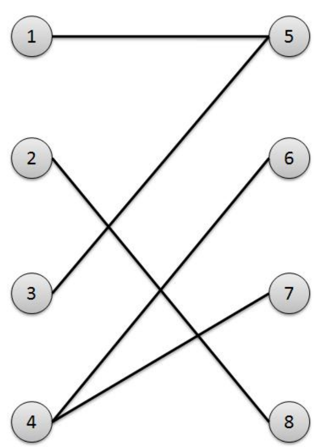

Figure 8: Graph generated after the dual feasible point is computed.

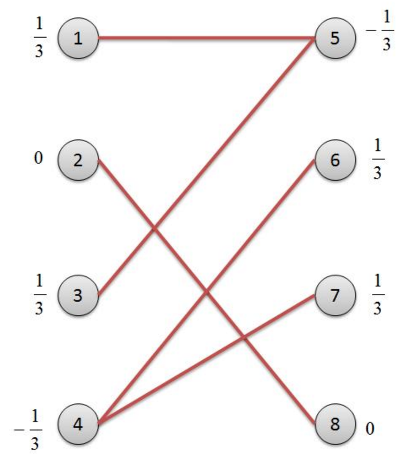

Figure 9: NNLS solution.

$-\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	
0	$6-\frac{2t}{3}$	$4-\frac{2t}{3}$	$2-\frac{t}{3}$	$\frac{1}{3}$
$5+\frac{t}{3}$	$2-\frac{t}{3}$	$3-\frac{t}{3}$	0	0
0	$4-\frac{2t}{3}$	$4-\frac{2t}{3}$	$2-\frac{t}{3}$	$\frac{1}{3}$
$4+\frac{2t}{3}$	0	0	$4+\frac{t}{3}$	$-\frac{1}{3}$

Figure 10: Maximum value of t such that dual feasibility will still be satisfied.

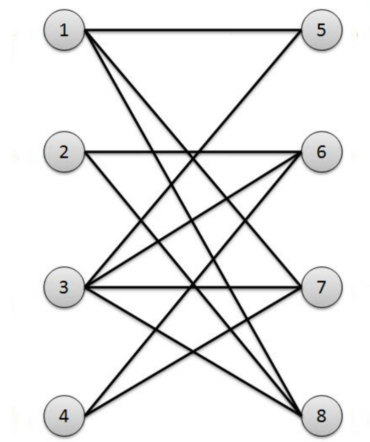

Figure 11: Graph generated after the dual feasible point is computed.

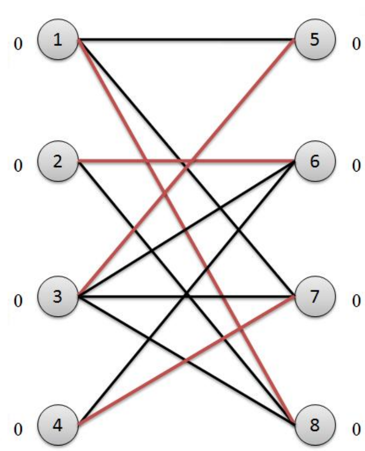

Figure 12: A possible final basis for the NNLS solution.

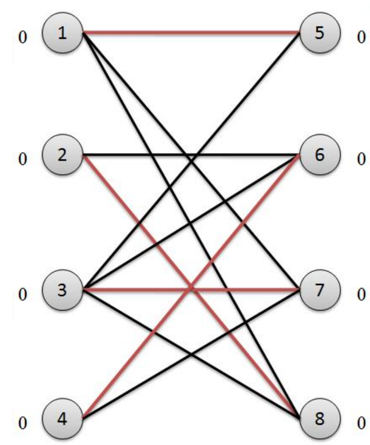

Figure 13: A possible final basis for the NNLS solution.

CHAPTER IV

SOLVING THE MAXIMUM CARDINALITY MATCHING PROBLEM USING THE NNLS ALGORITHM

Although the matching problem is very well understood and efficient algorithms for it on both bipartite and non-bipartite case exist, we aim at building even faster algorithms in practice for the general matching problem.

The main idea is to make a connection between solutions obtained by the NNLS algorithm and the solutions of the maximum cardinality matching problem for the same graph.

In section 4.1, we present a full characterization of the solutions obtained by the NNLS algorithm for both bipartite and non-bipartite graphs.

In section 4.2, look into the relationship between solutions obtained by the NNLS algorithm and the solutions of the maximum cardinality matching problem on general graphs.

4.1 Characterization of Solutions Obtained by the NNLS Algorithm

In this section we characterize the solution obtained by the NNLS algorithm.

4.1.1 The Bipartite Case

Let G be the input graph for the algorithm. We will assume throughout this subsection that G is bipartite. Each connected component of the solution obtained by the NNLS algorithm is a tree (see chapter 3) for the bipartite case. Thus, the solution obtained by the algorithm will be a forest. We will now characterize the arcs connecting these trees.

The problem can be formulated as:

$$(P): \min \quad \tfrac{1}{2}\sum_{j=1}^{n} s_j^2$$

$$\text{s.t.}$$

$$Ax + s = b$$

$$x \geq 0$$

Recall that the KKT conditions state that a vector (x, s) is a solution for P if and only if there exists a vector π such that:

$$(D): \quad \pi^t A_j = 0 \quad \text{if } x_j > 0 \tag{21}$$

$$\pi^t A_j \leq 0 \quad \forall j$$

$$\pi_j = s_j \quad \forall j$$

Let H_1 and H_2 be two connected components obtained by the NNLS algorithm. Let Δ_i and Ξ_i, with $|\Delta_i| \geq |\Xi_i|$, be the partitions of H_i, for $i = 1, 2$. Observe that Δ_i and Ξ_i are well defined for $i = 1, 2$ since G is bipartite. From chapter 3, we have that the unique solution value for the dual variables is:

$$\pi_k^i = \tfrac{|\Delta_i| - |\Xi_i|}{|H_i|} \quad \forall k \in \Delta_i \ , \ i = 1, 2.$$

$$\pi_k^i = \tfrac{|\Xi_i| - |\Delta_i|}{|H_i|} \quad \forall k \in \Xi_i \ , \ i = 1, 2.$$

Constraint (21) implies that $|\pi_k^i| = |\pi_j^i|$, $\forall k, j \in H_i$. Therefore, for each connected component H_i, let us define:

$$\pi^i = |\pi_j^i| \ , \quad \forall k \in H_i.$$

Thus, if an edge (u, v) ($u \in H_1$ and $v \in H_2$) connects the connected components H_1

and H_2. If $u \in H_1 \cap \Delta_1$ and $v \in H_2 \cap \Xi_2$, then

$$\begin{aligned}
\pi_u^1 + \pi_v^2 &\leq 0 \\
\pi_v^1 &\leq -\pi_u^2 \\
\pi^1 = |\pi_v^1| = \pi_v^1 &\leq -\pi_u^2 = |\pi_u^2| = \pi^2 \\
\pi^1 = |\pi_u^1| &\leq |\pi_v^2| = \pi^2
\end{aligned}$$

Thus, there can be an edge connecting $u \in H_1 \cap \Delta_1$ and $v \in H_2 \cap \Xi_2$ only if

$$\pi^1 \leq \pi^2$$

If $u \in H_1 \cap \Xi_1$ and $v \in H_2 \cap \Xi_2$, then there is no constraint, since

$$\pi_u^1 = -|\pi^1| < 0$$

$$\pi_v^2 = -|\pi^2| < 0$$

$$\pi_u^1 + \pi_v^2 = -|\pi^1| - |\pi^2| \leq 0$$

Thus, if $u \in H_1 \cap \Xi_1$ and $v \in H_2 \cap \Xi_2$, then

$$\pi_u^1 + \pi_v^2 \leq 0$$

is always satisfied.

It is clear that there can be no edge connecting $u \in H_1 \cap \Delta_1$ and $v \in H_2 \cap \Delta_2$, since:

$$\pi_u^1 = |\pi^1| \geq 0$$

$$\pi_v^2 = |\pi^2| \geq 0$$

$$\pi_u^1 + \pi_v^2 = |\pi^1| + |\pi^2| \geq 0$$

Thus, if one of the dual variables is nonzero, then we would have

$$\pi_u^1 + \pi_v^2 > 0$$

contradicting optimality.

It must be pointed out the we are characterizing the solutions obtained by the NNLS algorithm, and not general solutions for the problem. Not every solution for the minimization of the norm 2 of the vector of slack variables must be a tree. Consider the following example shown on figure 14 on four vertices (cycle of size four). The unique least squares solution is $\pi_1 = \pi_2 = \pi_3 = \pi_4 = 0$. However, for the values of x's (edges), more than one solution is possible. The red lines represent the edges that have positive value. The solution shown on figure 14(a), i.e., $x_{12} = x_{23} = x_{34} = x_{41} = 0.5$, is a possible assignment of value to the edges in the optimal solution, although it would have never been obtained by the NNLS algorithms. The solution on figure 14(b) ($x_{12} = x_{34} = 1$ and $x_{23} = x_{41} = 0$) could have been obtained by the algorithm, as well as the solution $x_{12} = x_{34} = 0$ and $x_{23} = x_{41} = 1$.

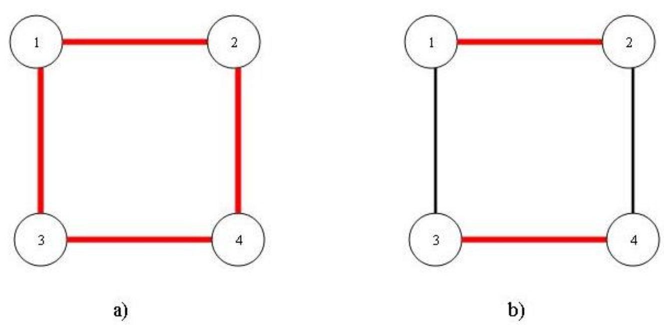

Figure 14: NNLS solution: possible solution values of x for the same graph.

4.1.2 The Non-bipartite Case

Let G be the input graph for the algorithm. We know that trees can be connected components of the solution of the NNLS algorithm. We will now show that an odd cycle is the only other type of connected component that will appear in a solution of the NNLS algorithm.

Lemma 12. *Let B be a feasible basis for the NNLS solution and I_B be its index set. Let $H \in G$ be a connected component and let $n = |H|$. Suppose, without loss of generality, that H spans G. If H is not a tree, then H must be an odd cycle.*

Proof. If B is a basis, then the columns of B must be linearly independent. If H is not a tree and it is a connected component, then it must have a cycle. Because of the linearly independence requirement of the columns, this cycle cannot be even. Therefore, H contains an odd cycle. We shall see that H is in fact an odd cycle. Suppose for a contradiction that H is composed of an odd cycle C connected to a tree T, i.e., $H = C \cup T$. Let $u \in T$ be a leaf. Since u is a leaf, $\pi_u = 0$. Suppose that u is connected to v. Since $\pi_u = 0$ then we must have $x_{uv} = 1$, and also that $\pi_v = 0$. This implies that the edge (u, v) is disconnected from C. Contradiction, since H is a connected subgraph of G. □

Again, it must be pointed out the we are characterizing the solutions obtained by the NNLS algorithm, and not general solutions for the least squares minimization problem. Not every solution for the minimization of the norm 2 of the vector of slack variables must be either a tree or an odd cycle. Consider the following example shown on figure 15 on five vertices (a clique of size five). The unique least squares solution is $\pi_1 = \pi_2 = \pi_3 = \pi_4 = \pi_5 = 0$. However, for the values of x's (edges), more than one solution is possible. The red lines represent the edges that have positive value. The solution shown on figure 15(a), i.e., $x_{12} = x_{13} = x_{14} = x_{15} = x_{23} = x_{24} = x_{25} = x_{34} = x_{35} = x_{45} = \frac{1}{4}$, is a possible assignment of value to the edges in the optimal solution, although it would have never been obtained by the NNLS algorithm. The solution on figure 15(b) ($x_{12} = x_{23} = x_{34} = x_{45} = x_{51} = \frac{1}{2}$ and $x_{13} = x_{14} = x_{24} = x_{25} = x_{35} = 0$) could have been obtained by the algorithm, as well as any the solution that contains an odd cycle.

Now that we have a full characterization of the connected components, we will characterize the edges connecting these connected components. So far, in subsection

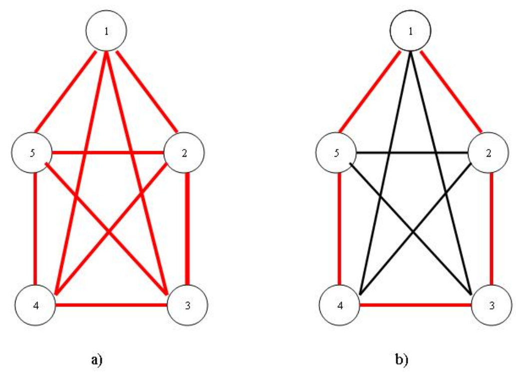

Figure 15: NNLS solution: possible solution values of x for the same graph.

4.1.2, we characterized the value of the dual variables corresponding to arcs that connect two trees.

Since the dual variables corresponding to the arcs are zero, the reduced cost of any arc connecting two cycles is zero, thus there is no restriction in this case.

Now suppose let an edge (u, v) connect the tree H to a cycle C.

$$\begin{aligned} \pi_u^H + \pi_v^C &\leq 0 \\ \text{Since} \quad \pi_v^C = 0 \quad \Rightarrow \quad \pi_u^H &\leq 0 \end{aligned}$$

Thus, any connected component H in the solution obtained by the NNLS that is a tree will be connected to cycle by an edge incident with a vertex $u \in H$ only if $\pi_u^H \leq 0$, that is, if u belongs to the partition of smaller cardinality of H_1.

4.2 Relationship between the Maximum Matching and NNLS Solutions

In this section, we discuss in detail the relationship between solutions obtained by the NNLS algorithm and the solutions of the maximum cardinality matching problem on general graphs.

4.2.1 The Bipartite Case

Let G be the input graph for the algorithm. We will assume throughout this subsection that G is bipartite. Also, the partitions of G will be denoted as Δ and Ξ, and these are such that $|\Delta| \geq |\Xi|$.

In section 4.1 it was stated that if S is bipartite, the solution obtained by the NNLS algorithm is necessarily a forest. The next theorem gives a necessary condition for a connected component in the NNLS solution in terms of the maximum cardinality matching of that component.

Theorem 13. *Let $H \in G$ be a connected component obtained by the NNLS algorithm. Let Δ and Ξ be the partitions of H such that $|\Delta| \geq |\Xi|$. Then the maximum cardinality matching for H leaves $|\Delta| - |\Xi|$ exposed nodes.*

Proof. Applying Cauchy-Schwartz inequality to a vector $s \in R^n$ we have that:

$$\sum_{j=1}^{n} s_j \leq \sqrt{\sum_{j=1}^{n} s_j^2}\sqrt{n} \tag{22}$$

Thus:

$$\begin{aligned} n - 2\sum_{j=1}^{n} x_j = \sum_{j=1}^{n} s_j \quad &\leq \quad \sqrt{\sum_{j=1}^{n} s_j^2}\sqrt{n} \\ n - 2\sum_{j=1}^{n} x_j \quad &\leq \quad \sqrt{\sum_{j=1}^{n} s_j^2}\sqrt{n} \end{aligned}$$

Now, let z_p be the solution value of the following problem:

$$\begin{aligned} (P) : \min \quad & \textstyle\sum_{j=1}^{n} s_j^2 \\ \text{s.t.} \quad & \\ & Ax + s = b \\ & x \geq 0 \end{aligned}$$

Let z_q be the solution value of the following problem:

$$\begin{aligned} (Q) : \min \quad & \textstyle\sum_{j=1}^{n} s_j \\ \text{s.t.} \quad & \\ & Ax + s = b \\ & x \geq 0 \end{aligned}$$

From (22), we have

$$z_q \leq \sqrt{z_p}\sqrt{n}$$

Finally, let z_{q^+} be the solution value of the following problem, that is equivalent to the maximum cardinality matching problem:

$$\begin{aligned} (Q^+) : \min \quad & \textstyle\sum_{j=1}^{n} s_j \\ \text{s.t.} \quad & \\ & Ax + s = b \\ & x \geq 0 \\ & s \geq 0 \end{aligned}$$

It is clear that:

$$z_{q^+} = z_q$$

Therefore:

$$z_{q^+} \leq \sqrt{z_p}\sqrt{n} \tag{23}$$

From (23), it is clear that z_{q^+} is the number of exposed nodes on the maximum cardinality matching. Now, note that the solution value of (P) for the connected component H will be

$$z_p = \sum_{j=1}^{n} s_j^2 = \sum_{j=1}^{n} \frac{(|\Delta| - |\Xi|)^2}{n^2} = n\frac{(|\Delta| - |\Xi|)^2}{n^2} = \frac{(|\Delta| - |\Xi|)^2}{n} \tag{24}$$

Thus, substituting (24) in (23):

$$z_{q^+} \leq \sqrt{\frac{(|\Delta| - |\Xi|)^2}{n}}\sqrt{n}$$
$$z_{q^+} \leq \frac{|\Delta| - |\Xi|}{\sqrt{n}}\sqrt{n}$$
$$z_{q^+} \leq |\Delta| - |\Xi| \tag{25}$$

Thus, inequality (25) gives an upper bound on the number of exposed nodes. Since it is clear that $z_{q^+} \geq |\Delta| - |\Xi|$, then the number of exposed nodes in the maximum cardinality matching for the component H must be exactly $|\Delta| - |\Xi|$. □

To illustrate an application of theorem 13, consider the graph on 7 vertices shown in figure 16. The partitions are $\Delta = \{1, 2, 4, 6\}$ and $\Xi = \{3, 5, 7\}$. On red we can see a maximum cardinality matching. The number of exposed nodes is $7 - 4 = 3$. Now, since $3 > 1 = |\Delta| - |\Xi|$, the theorem implies that the NNLS solution for G has to have more than 1 connected component, as shown in figure 17. The connected components satisfy theorem 13, as we can see on figures 18 and 19.

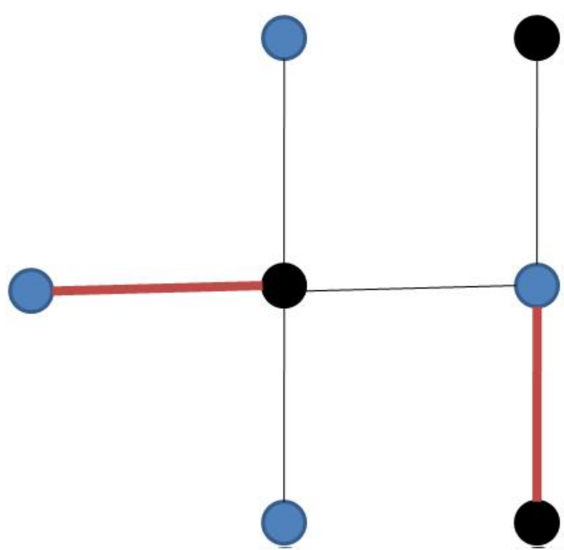

Figure 16: Maximum matching on a graph that has more than 1 connected component in the NNLS solution.

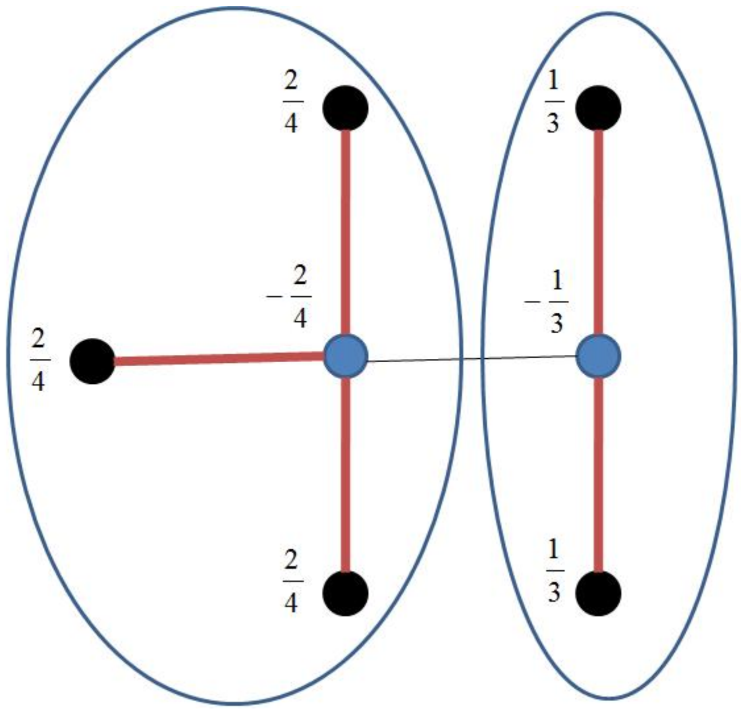

Figure 17: NNLS solution for the graph on figure 16.

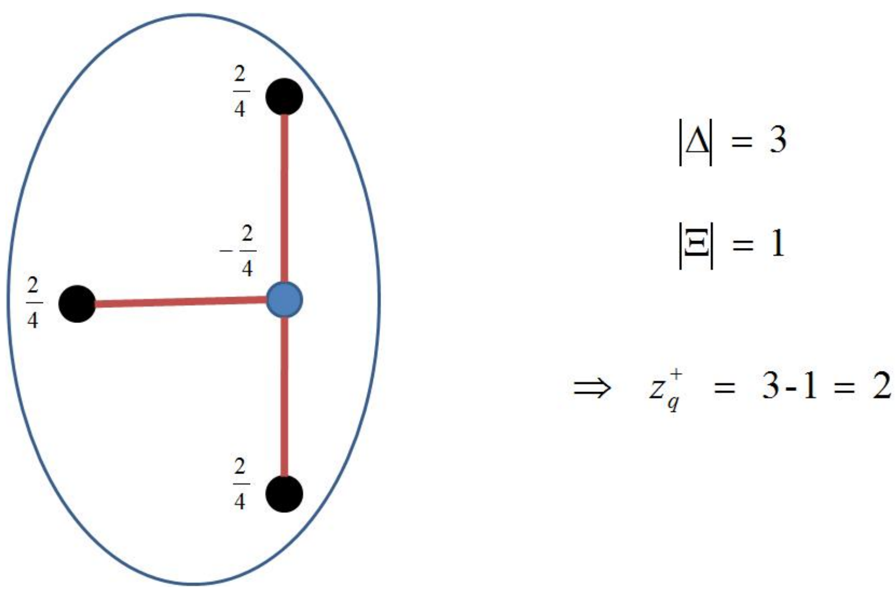

Figure 18: Connected component of the NNLS solution in figure 17.

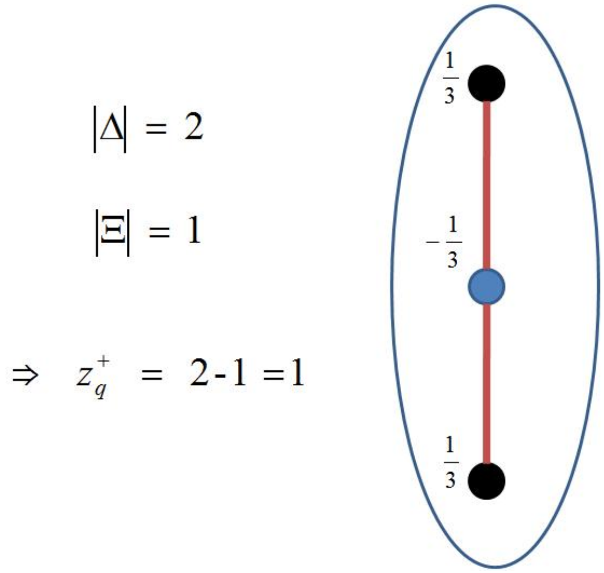

Figure 19: Connected component of the NNLS solution in figure 17.

Corollary 14. *Let $H \in G$ be a connected component obtained by the NNLS algorithm. Let Δ and Ξ be the partitions of H such that $|\Delta| \geq |\Xi|$. Then there exists a complete matching from Ξ to Δ.*

Proof. From theorem 13, we know that the maximum matching for H leaves $|\Delta| - |\Xi|$ exposed nodes. Thus, its cardinality must be $|\Xi|$. Since there can be no edge connecting two vertices in $|\Delta|$ (since H is bipartite), there are no exposed nodes in $|\Xi|$. □

Let π^* be the solution of problem (P) in section 4.1.2. Let us define the following sets:

$$J^0 = \{j|\pi_j^* = 0\} \tag{26}$$

$$J^- = \{j|\pi_j^* < 0\} \tag{27}$$

$$J^+ = \{j|\pi_j^* > 0\} \tag{28}$$

Theorem 15. *Let G be a bipartite graph with partition (Δ, Ξ) and A its incidence matrix. Further, let π^* be the solution of problem (P) in section 4.1.2 and consider the sets J^0, J^-, J^+ defined previously. Let $M^- = \{(u,v)|u \in J^- \text{ , } v \in J^+\}$ such that M^-*

is a matching and $|M^-| = |J^-|$ *and* $M^0 = \{(u,v)|u \in J^0 \cap \Xi\}$*. The* $M = M^- \cup M^0$ *is a maximum cardinality matching on* G*.*

Proof. Firstly, it must be observed that the existence of a matching M^- with the aforementioned properties in the statement of the theorem follows from corollary 14.

Also, since M^0 is clearly seen to be a perfect matching on the graph induced by the vertices in J^0, M^0 could have been constructed also with the set Δ, i.e., $M^0 = \{(u,v)|u \in J^0 \cap \Delta\}$.

Clearly M is a matching, otherwise the graph would contain an edge connecting two vertices on the same partition. We need to show that there is no augmenting path in G for M.

Suppose for a contradiction that there is an augmenting path P. By construction of M, the exposed nodes must belong to J^+. Suppose that P connects $u, v \in P \cap J^+$. Clearly, u cannot be connected to any vertex in M^0, otherwise it would violate the KKT conditions and π^* would not be a solution. That means that the path P connecting u to v must contain edges belonging to M^-. But since this path is alternating, there will be a vertex in J^+ connected to u or v, and this is a contradiction, since it will violate the KKT conditions. □

Theorem 16. *Let* G *be a bipartite graph with partition* (Δ, Ξ) *and* A *its incidence matrix. Further, let* π^* *be the solution of problem* (P) *in section 4.1.2 and consider the sets* J^0, J^-, J^+ *defined previously. Let* $C^- = \{u|u \in J^-\}$ *and* $C^0 = \{u|u \in J^0 \cap \Xi\}$*. The* $C = C^- \cup C^0$ *is a minimum cardinality cover on* G*.*

Proof. We have to show first that C is a cover of the edges by vertices. Suppose that C is not a cover. Then there must be an edge connecting a vertex $u \in J^+$ to a vertex $v \in C^0 \cup J^+$. But then KKT conditions would be violated, since by construction $\pi_u + \pi_v > 0$.

Clearly there cannot be a vertex cover with fewer vertices. □

Corollary 17. (König's theorem) *In any bipartite graph, the number of edges in a maximum cardinality matching is equal to the minimum cardinality vertex cover.*

Proof. Take the matching M as in theorem 15 and the vertex cover C as in theorem 16. Clearly $|M| = |C|$. □

4.2.1.1 An Example

To illustrate theorems 15 and 16, consider the bipartite graph on figure 20. One partition has the blue color and the other is black.

On figure 21 we can see the NNLS solution. From the solution shown on figure 21, we can construct the sets J^0, J^-, J^+ and using theorems 15 and 16 obtain the maximum matching on figure 22 as well as the minimum cover on figure 23.

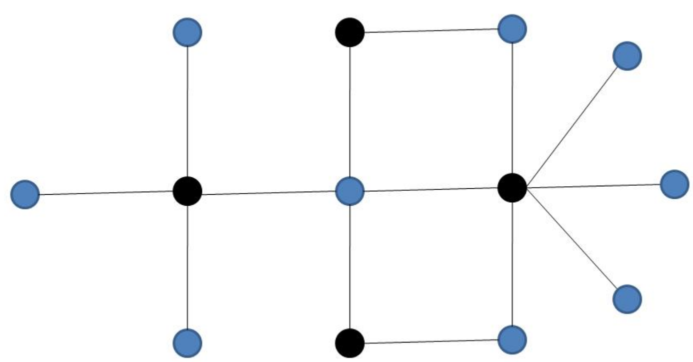

Figure 20: A bipartite graph.

4.2.2 The Non-Bipartite Case

For the non-bipartite case the we do not have the nice connection between the NNLS solution and the maximum cardinality matching provided by theorem 15. The reason is that odd cycles cause the NNLS to produce solutions such that some of the

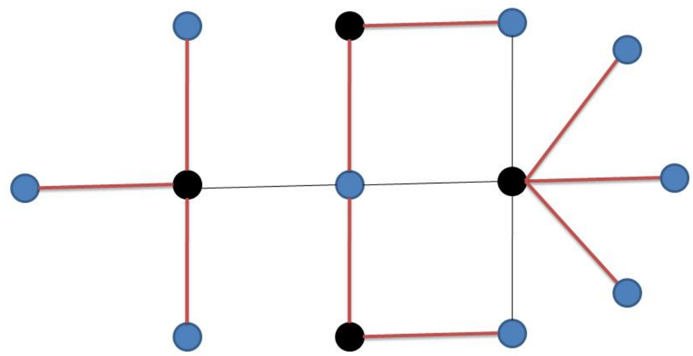

Figure 21: NNLS solution.

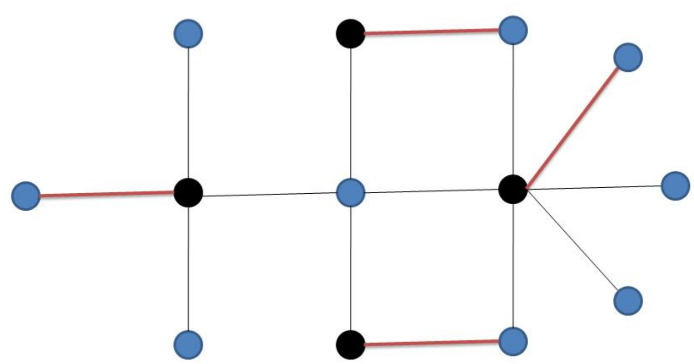

Figure 22: Maximum matching.

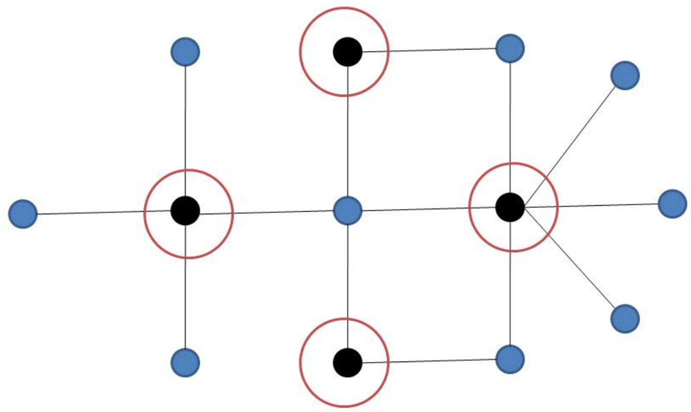

Figure 23: Minimum cover.

basic variables do not belong to any maximum matching. To illustrate this, consider the graph on figure 24.

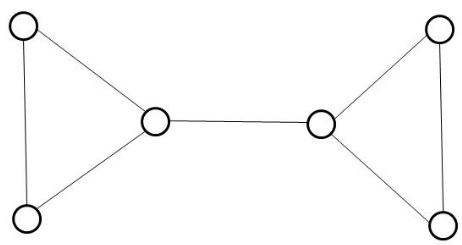

Figure 24: Example of a non-bipartite graph.

On figure 25 we can see the unique maximum matching. However, the NNLS will not necessarily produce this matching as a final basis. One possible basis for the NNLS algorithm is the one highlighted on figure 26, leaving out one edge of the maximum matching.

In order the avoid the situation illustrated on figure 26, we need to find a way of making the NNLS algorithm give us a basis with the fewest number of odd cycles.

In order to achieve this, we will have to generate a new graph G^* induced by the

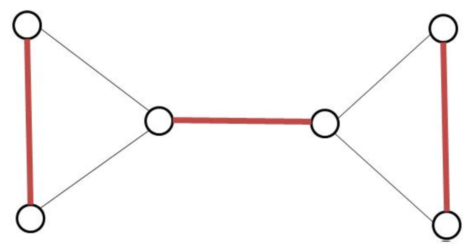

Figure 25: Unique maximum matching.

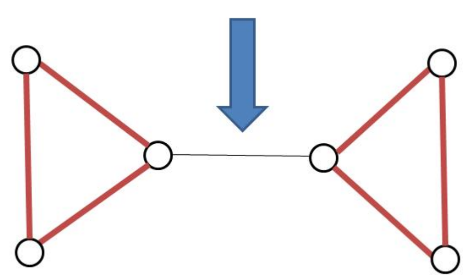

Figure 26: Possible NNLS solution.

vertices whose dual variable are zero. Thus, $V(G^*) = J^0$. It is clear that if we apply the NNLS algorithm to G^*, the answer will be the same and the same cycle may be generated. Therefore, we need a way to prevent this cycles to be in the solution. In order to accomplish that, we resort to shrinking each odd cycle (an idea developed by Edmonds in [12]) into a node and apply the NNLS in the smaller graph and apply the same idea recursively. One crucial difference between this algorithm and the traditional ones in the literature that use the blossom idea (see [15], [25], [34] and [16]) is that several odd cycles are shrunk at the same time, since the solution of the NNLS will generate all of them after the execution.

It has been shown in section 4.1 that the solutions generated by the NNLS algorithm will be a graph (possibly disconnected) composed of connected components that are trees (see chapter 3) or odd cycles (see lemma 12). Also, from lemma 12, we know that the odd cycles in E^- are isolated.

In order for us to build a procedure to extract the maximum matching from the NNLS solution for a general graph, we need the following lemma:

Lemma 18. *Let G be a general graph. Suppose that we start the NNLS with a basis that is a forest (e.g. any matching on G). Let (x^*, π^*) be the NNLS solution and let B be the optimal basis. If B has the smallest number of odd cycle, then the maximum matching will be contained in the columns of the basis.*

Proof. Consider the linear programming formulation of the maximum matching problem on a graph with n nodes is:

$$
\begin{aligned}
(P):\max \quad & \textstyle\sum_{e\in E(G)} x_e \\
\text{s.t.} \quad & \\
& Ax + s = 1 \\
& x \geq 0
\end{aligned}
$$

Summing up the equality constraint of problem (P):

$$2\sum_{e\in E(G)} x_e + \sum_{j\in V(G)} s_j = n$$

Therefore, $\max\sum_{e\in E(G)} x_e$ is equivalent to $\min\sum_{j\in V(G)} s_j$. We know that every odd cycle produced by the NNLS solution is isolated. Therefore, each odd cycle will have one exposed node. Thus, minimizing the number of odd cycles will minimize the number of exposed nodes in the matching obtained from the NNLS solution. □

Lemma 18 states that if we can devise a procedure to compute the NNLS solution such that the optimal basis will have the fewest number of odd cycles, then we can extract the maximum cardinality matching from it. This is crucial in the construction of an algorithm that uses the NNLS procedure to compute the maximum matching, since if we simply calculate the maximum matching for each of the cycles obtained, it is easy to see that there may still be an augmenting path connecting two exposed nodes in these cycles. Therefore, after obtaining the NNLS solution for G, we have a maximum matching on $J^- \cup J^+$ that will be part of the maximum matching on G, we can eliminate the subgraph induced by vertices in $J^- \cup J^+$ and rerun the NNLS algorithm for the subgraph of G induced by the vertices in J^0 obtained after shrinking the odd cycles.

4.2.2.1 An Example

To illustrate how the NNLS can be used to compute the maximum cardinality matching by means of lemma 18, consider the non-bipartite graph on figure 27.

Suppose that after applying the NNLS algorithm we end up with the basis on figure 28. The edges that are in the basis are in red. The unique value of the dual variables is indicated next to each node. The vertices in J^- are in blue and the ones in J^+ are in black.

Figure 29 is the graph that remains when we remove all of the vertices in $J^- \cup J^+$. Also the odd cycles remain in red as to indicate which odd cycle will shrink.

After shrinking the odd cycle on figure 29, we obtain the graph on figure 30. Again the NNLS algorithm is applied to this reduced graph, obtaining the solution on figure 31.

The odd cycles in the basis of the new solution are highlighted on figure 32. After shrinking (see figure 33) and solving the NNLS for the new reduced graph. we have the solution shown on figure 34. Since there are no longer odd cycles in the last basis obtained, the algorithm stops.

On figure 35 and 36 we can see the expansion of the odd cycle taking place in order to recover the matching for the original graph shown on figure 37.

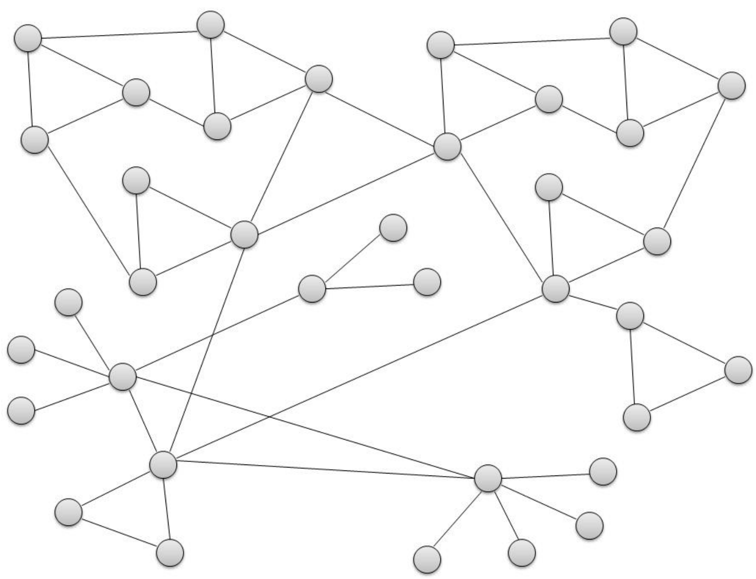

Figure 27: Example of the NNLS algorithm applied to non-bipartite graphs.

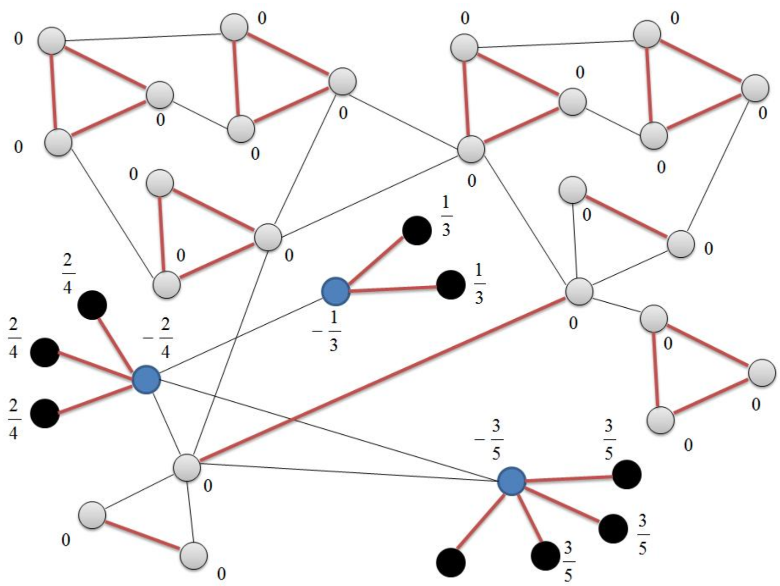

Figure 28: NNLS solution.

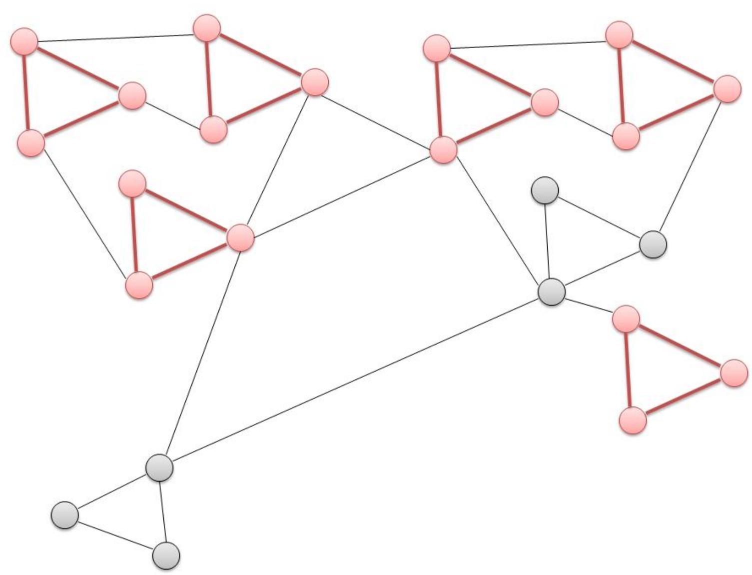

Figure 29: Odd cycles in the optimal basis.

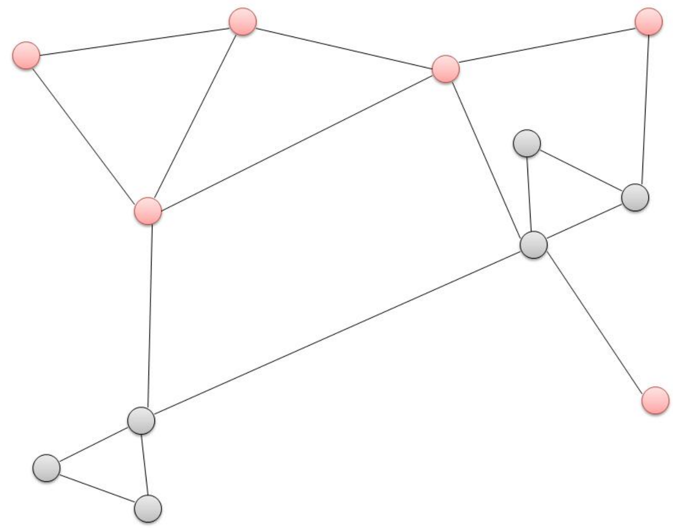

Figure 30: Graph obtained after shrinking the odd cycles.

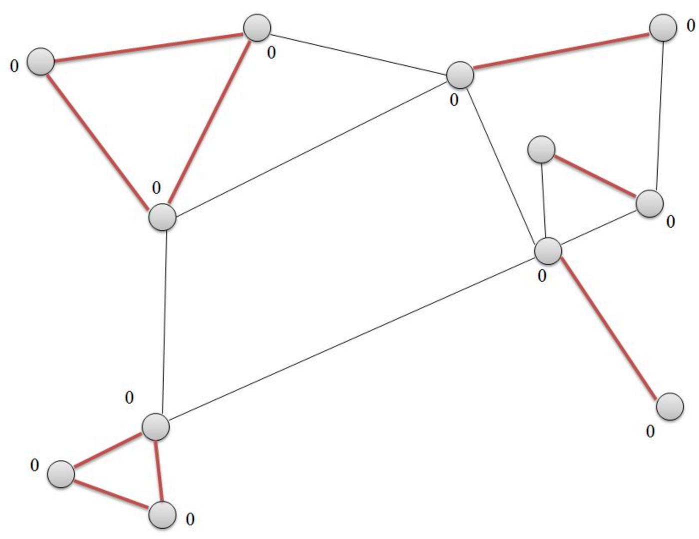

Figure 31: New NNLS for the reduced graph.

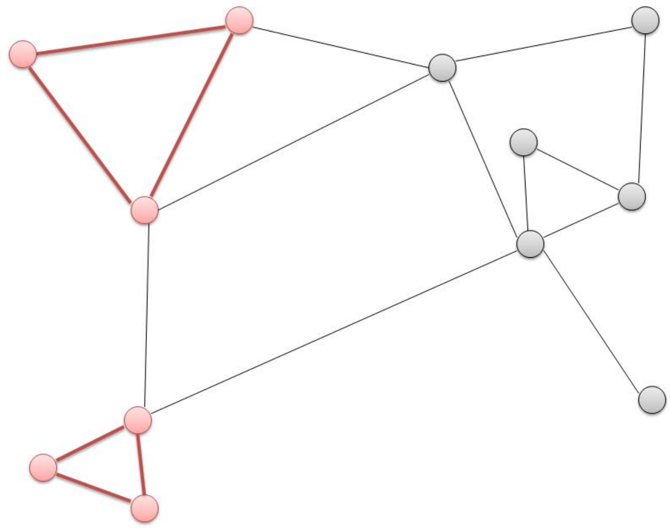

Figure 32: Odd cycles in the optimal basis.

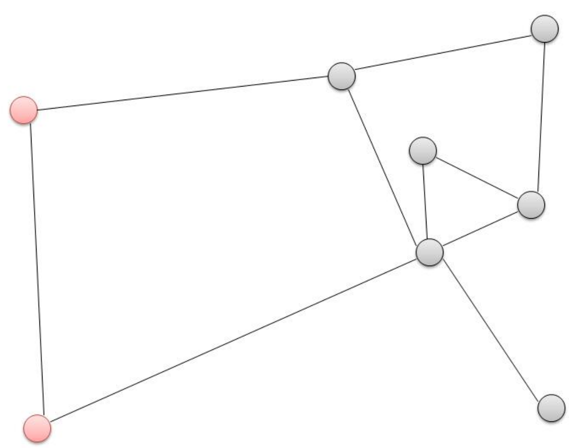

Figure 33: Graph obtained after shrinking the odd cycles.

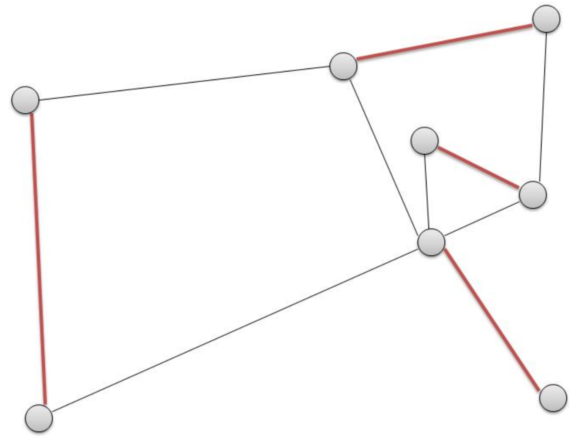

Figure 34: New NNLS for the reduced graph.

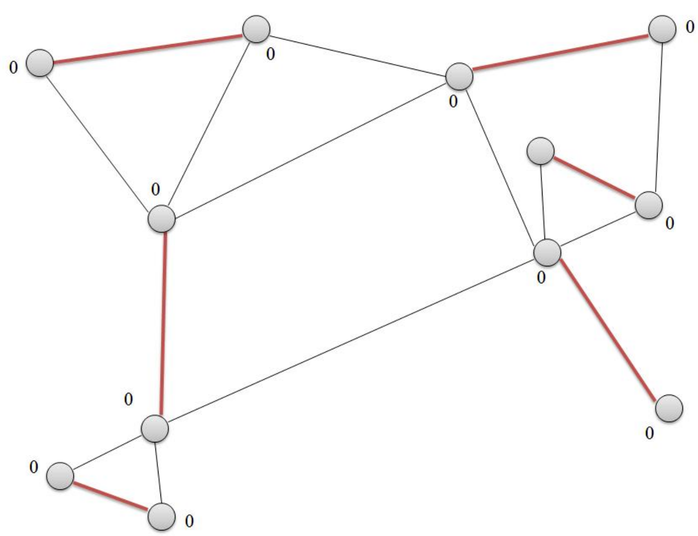

Figure 35: Expanding the odd cycles.

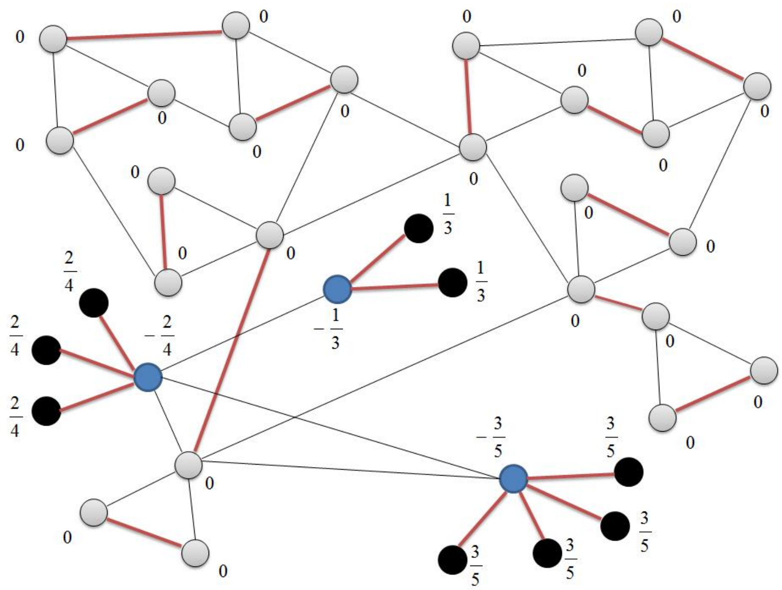

Figure 36: New basis of the NNLS with the minimum number of odd cycles.

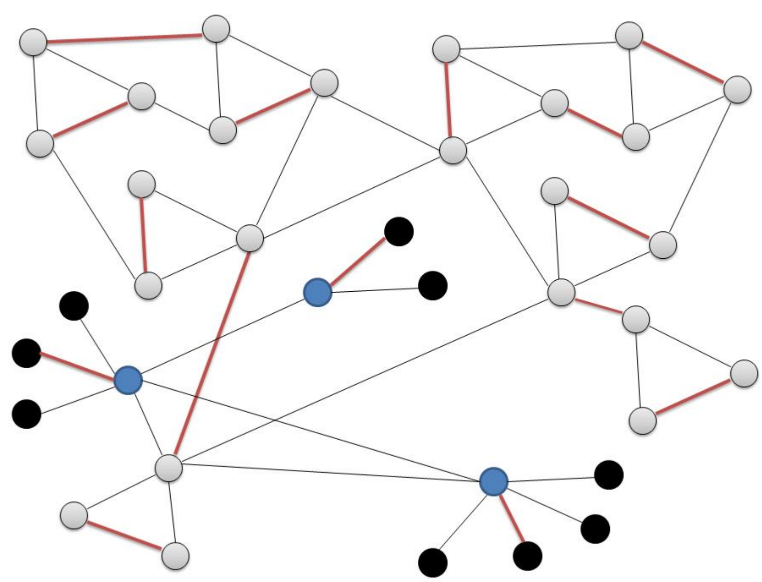

Figure 37: Maximum matching.

CHAPTER V

MINIMIZATION OF P-NORMS AND SEPARABLE DIFFERENTIABLE CONVEX FUNCTIONS

In this chapter, we describe some theoretical results concerning the solution of minimization of p-norms and separable differentiable convex functions subject to matrices with a special structure: each column must have at most two nonzero elements.

In section 5.1, we reduce the general problem of minimizing separable differentiable convex functions to the minimization in norm 2 allowing us to use the NNLS algorithm to solve it.

In section 5.2, we deal with the infinity norm. Since the infinity norm is not differentiable, we must use an approach different from the one used in section 5.1.

5.1 Separable Differentiable Convex Functions

Let A be a $m \times n$ matrix and b be a vector of dimension m. Let also $f(.)$ be a convex differentiable functions. We are interested in solving the following problem:

$$
\begin{aligned}
(P) : \min \quad & \textstyle\sum_{j=1}^{n} f(s_j) \\
\text{s.t.} \quad & \\
& Ax + s = b \\
& x \geq 0
\end{aligned}
$$

Since P is a convex program, KKT are necessary and sufficient. Thus, the vector

(x, s) is a solution for P if and only if there exists π such that:

$$
\begin{aligned}
(D): \quad & \pi^t A_j = 0 \quad \text{if } x_j > 0 \\
& \pi^t A_j \leq 0 \quad \forall j \\
& \pi_j = f'(s_j) \quad \forall j
\end{aligned}
$$

Suppose that the matrix A has exactly two nonzero elements per column. For a column j, it will be denoted by j_1 and j_2 the nonzero elements of this column. Using this notation, the KKT conditions are written as:

$$
\begin{aligned}
(D): \quad & \pi_{j_1} A_{j_1} + \pi_{j_2} A_{j_2} = 0 \quad \text{if } x_j > 0 \\
& \pi_{j_1} A_{j_1} + \pi_{j_2} A_{j_2} \leq 0 \quad \forall j \\
& \pi_j = f'(s_j) \qquad \forall j
\end{aligned}
$$

Substituting $\pi_j = f'(s_j)$ on the other inequalities:

$$
\begin{aligned}
(D): \quad & f'(s_{j_1}) A_{j_1} + f'(s_{j_2}) A_{j_2} = 0 \quad \text{if } x_j > 0 \\
& f'(s_{j_1}) A_{j_1} + f'(s_{j_2}) A_{j_2} \leq 0 \quad \forall j
\end{aligned}
$$

This in turn is:

$$
\begin{aligned}
(D): \quad & f'(s_{j_1}) A_{j_1} = -f'(s_{j_2}) A_{j_2} \quad \text{if } x_j > 0 \\
& f'(s_{j_1}) A_{j_1} \leq -f'(s_{j_2}) A_{j_2} \quad \forall j
\end{aligned}
$$

Let us first consider the simplest case where A is a $\{0, 1\}$ matrix. Then the previous formula reduces to:

$$
f'(s_{j_1}) = -f'(s_{j_2}) \quad \text{if } x_j > 0 \tag{29}
$$

$$
f'(s_{j_1}) \leq -f'(s_{j_2}) \quad \forall j \tag{30}
$$

Suppose now that we want to solve problem (P) for another differentiable convex function $g(.)$.

$$(P_g): \min \quad \sum_{j=1}^{n} g(s_j)$$

$$\text{s.t.}$$

$$Ax + s = b$$

$$x \geq 0$$

$$g'(s_{j_1}) = -g'(s_{j_2}) \quad \text{if } x_j > 0 \tag{31}$$

$$g'(s_{j_1}) \leq -g'(s_{j_2}) \quad \forall j \tag{32}$$

Suppose, further, that we have the solution for (P). A natural question arises: for a constraint matrix with the state properties, what are the necessary and sufficient conditions for the problem (P_g) has the same solution as (P)?

Clearly, if the function $g(.)$ on (P_g) satisfies 29 and 30 are true if and only if 31 and 32 hold, then a solution of (P) is also a solution of (P_g), and vice-versa, since the KKT conditions, in this case, are necessary and sufficient. From this discussion, we have the following lemma:

Lemma 19. *Let A be a $\{0,1\}$ matrix with exactly two nonzero elements in each column. Let x be the solution of (P). Let $g(.)$ be a convex differentiable function such that*

$$g'(s_{j_1}) = -g'(s_{j_2}) \quad \Leftrightarrow \quad f'(s_{j_1}) = -f'(s_{j_2}) \tag{33}$$

$$g'(s_{j_1}) \leq -g'(s_{j_2}) \quad \Leftrightarrow \quad f'(s_{j_1}) \leq -f'(s_{j_2}) \tag{34}$$

Then x is also a solution of (P_g).

Proof. This can be clearly inferred from the KKT conditions for the problems (P) and (P_g). □

Theorem 20. *Let A be a $\{0,1\}$ matrix with exactly two nonzero elements in each column. Let $f(x) = \frac{1}{2}x^2$. Let x be the solution of (P). Let $g(.)$ be a convex differentiable function such that $g'(.)$ is an injective odd function. Then x is also a solution of (P_g).*

Proof. If suffices to show that equivalences 33 and 34 hold for any x. Since the matrix is $\{0,1\}$ and there are exactly two nonzero elements in each column, if s is a solution for (P), then:

$$f'(s_{j_1}) = -f'(s_{j_2}) \Leftrightarrow s_{j_1} = -s_{j_2} \Leftrightarrow g(s_{j_1}) = g(-s_{j_2}) \Leftrightarrow g'(s_{j_1}) = -g'(s_{j_2})$$

The above equivalences follow from the definition of $f(.)$ and from the fact that $g'(.)$ is an injective odd function. That proves equivalence 33.

Equivalence 34 can be proved using the same reasoning:

$$f'(s_{j_1}) \leq -f'(s_{j_2}) \Leftrightarrow s_{j_1} \leq -s_{j_2} \Leftrightarrow g(s_{j_1}) \leq g(-s_{j_2}) \Leftrightarrow g'(s_{j_1}) \leq -g'(s_{j_2})$$

The above equivalences follow from the definition of $f(.)$ and from the fact that $g'(.)$ is an injective odd function. □

Corollary 21. *Let A be a $\{0,1\}$ matrix with exactly two nonzero elements in each column. Let $f(x) = \frac{1}{2}x^2$. Let x be the solution of (P). Let $g(x) = \frac{1}{p}x^p$, for any even p, $p > 2$. Then x is also a solution of (P_g).*

Proof. Straightforward application of theorem 20, since $g^{'}(x) = x^{p-1}$ is an injective odd function. □

The previous corollary tells us that given a constraint $\{0,1\}$-matrix A with exactly two nonzero elements in each column, the solution of the NNLS problem is the same as for any p-norm, $p > 2$.

As an example, let us consider the general graph on figure 38. The norm 2 solution is on figure 39. The p-norm solution is shown on figure 40.

5.2 *The Infinity Norm*

In this section, we show how to obtain the solution for the infinity norm from the norm 2 solution.

In the previous section we dealt with norms that were differentiable functions. The infinity norm, however, is not differentiable, precluding us from using the same approach to solve this problem as for the case when p is finite.

For a column j, it will be denoted by j_1 and j_2 the nonzero elements of this column.

The minimization for the infinity norm can be posed as the following linear programming problem:

$$
\begin{aligned}
(P^\infty):\quad & \min \quad z \\
\text{s.t.} \quad & \\
& z - s_j^+ - s_j^- \geq 0 \quad \forall j \\
& Ax + s^+ - s^- = b \\
& x, s^+, s^- \geq 0
\end{aligned}
$$

The dual of (P^∞) is

$$
\begin{aligned}
(D^\infty):\quad & \max \quad \pi^t b \\
\text{s.t.} \quad & \\
& \textstyle\sum_j \rho_j = 1 \\
& \pi_j - \rho_j \leq 0 \quad \forall j \\
& \pi_j + \rho_j \geq 0 \quad \forall j \\
& \pi_{j_1} + \pi_{j_2} \leq 0 \quad \forall j \\
& \rho \geq 0
\end{aligned}
$$

Theorem 22. *Let A be a $\{0,1\}$ matrix with exactly two nonzero elements in each* *column. Then*

$$\min_{x,s} \|b - Ax\|_\infty = \max\{s_j | s_j \in \arg\min\{\|b - Ax\|^2\}\}$$

Proof. Let $(\hat{x}, \hat{s})$ be the solution for the norm 2 problem:

$$\begin{aligned} (P) : \min \quad & \textstyle\sum_{j=1}^{n} \frac{1}{2} s_j^2 \\ \text{s.t.} \quad & \\ & Ax + s = b \\ & x \geq 0 \end{aligned}$$

Also, let $\hat{\pi}$ be the solution of the dual of (P), i.e., $\hat{\pi} = \hat{s}$. We will show that the solution value of (P^∞) is $\max_j \hat{\pi}_j$.

Recall that for a connected component in the norm 2 solution, $|\hat{\pi}_i| = |\hat{\pi}_j|$, for any edge (i, j) in this component.

Let $\tilde{s} = \max_j \hat{s}_j$.

Consider now the connected components such that $\tilde{s} = |\hat{\pi}_j|$, for any vertex j of this connected component. Let us denote the vertex set of these components by $C \subset V(G)$. Set

$$\begin{aligned} \tilde{\rho}_j = & \quad \frac{|\pi_j|}{\sum_{k \in C} |\pi_k|} & \forall j \in C \\ \tilde{\rho}_j = & \quad 0 & \forall k \in V(G) - C. \end{aligned}$$

Also, set

$$\begin{aligned} \tilde{\pi}_j = & \quad \tilde{\rho}_j, \text{ if } \hat{\pi}_j > 0 & \forall j \in C \\ \tilde{\pi}_j = & \quad -\tilde{\rho}_j, \text{ if } \hat{\pi}_j < 0 & \forall j \in C \\ \tilde{\pi}_j = & \quad 0 & \forall k \in V(G) - C. \end{aligned}$$

Claim 23. *$(\tilde{\rho}, \tilde{\pi})$ is feasible for (D^∞).*

Proof. By construction, it is easy to see that all of the constraints except 35 are satisfied. Thus, let us check that constraint 35 is satisfied by $(\tilde{\rho}, \tilde{\pi})$. Suppose $j, k \in C$. Then

$$\tilde{\pi}_j + \tilde{\pi}_k = \tilde{\rho}_j - \tilde{\rho}_j = 0$$

Suppose $j, k \in V(G) - C$. Then

$$\tilde{\pi}_j + \tilde{\pi}_k = 0 - 0 = 0$$

Suppose $j \in C$ and $k \in V(G) - C$.

$$\tilde{\pi}_j + \tilde{\pi}_k = -\tilde{\rho}_j + 0 = -\tilde{\rho}_j \leq 0$$

The first equality holds, since otherwise

$$\tilde{\pi}_j = \tilde{\rho}_j > 0 \quad \Rightarrow \quad \hat{\pi}_j = \max_i \hat{s}_i \quad \Rightarrow \quad \hat{\pi}_j + \hat{\pi}_k > 0$$

Contradicting the dual optimality of $\hat{\pi}$. $\square$

Let $\tilde{z}_p$ be the solution value of P^∞ and $\tilde{z}_d$ be the solution value of D^∞.

Claim 24. $\tilde{s} = \tilde{z}_p = \tilde{z}_d$.

Proof. The first equality comes from the definition of $\tilde{s}$. We will prove now the second one by showing that the dual feasible solution $(\tilde{\rho}, \tilde{\pi})$ has value $\tilde{s}$.

$$\begin{aligned}
\tilde{z}_d \geq b^t\tilde{\pi} = & \quad \textstyle\sum_{j|\hat{\pi}_j>0} b_j\tilde{\pi}_j + \sum_{j|\hat{\pi}_j<0} b_j\tilde{\pi}_j \\
= & \quad \textstyle\sum_{j|\hat{\pi}_j>0} b_j\tilde{\rho}_j - \sum_{j|\hat{\pi}_j<0} b_j\tilde{\rho}_j \\
= & \quad \textstyle(\sum_{j|\hat{\pi}_j>0} b_j - \sum_{j|\hat{\pi}_j<0} b_j)\tilde{\rho}_j \\
= & \quad \textstyle(\sum_{j|\hat{\pi}_j>0} b_j - \sum_{j|\hat{\pi}_j<0} b_j)\frac{|\pi_j|}{\sum_{k\in C}|\pi_k|} \\
= & \quad \textstyle(\sum_{j|\hat{\pi}_j>0} b_j - \sum_{j|\hat{\pi}_j<0} b_j)\frac{\tilde{s}}{\sum_{k\in C}|\pi_k|}
\end{aligned}$$

Since $|\pi_k| = \frac{\sum_{j|\hat{\pi}_j>0} b_j - \sum_{j|\hat{\pi}_j<0} b_j}{|C|}$, then

$$\begin{aligned}
\sum_{k\in C} |\pi_k| &= |C|\frac{\sum_{j|\hat{\pi}_j>0} b_j - \sum_{j|\hat{\pi}_j<0} b_j}{|C|} \\
\sum_{k\in C} |\pi_k| &= \sum_{j|\hat{\pi}_j>0} b_j - \sum_{j|\hat{\pi}_j<0} b_j
\end{aligned}$$

Therefore

$$\begin{aligned}
\tilde{z}_d \geq & \left(\sum_{j|\hat{\pi}_j>0} b_j - \sum_{j|\hat{\pi}_j<0} b_j\right)\frac{\tilde{s}}{\sum_{k\in C} |\pi_k|} \\
= & \left(\sum_{j|\hat{\pi}_j>0} b_j - \sum_{j|\hat{\pi}_j<0} b_j\right)\frac{\tilde{s}}{\sum_{j|\hat{\pi}_j>0} b_j - \sum_{j|\hat{\pi}_j<0} b_j} \\
= & \tilde{s}
\end{aligned}$$

□

Thus, we found a dual feasible solution with the same value as the primal feasible obtained by the norm 2 solution, proving the theorem. □

As an example of how to extract the infinity norm solution from the NNLS solution, consider the general graph on figure 38. The NNLS solution can be seen on figure 39. From this norm 2 solution, we identify the connected component whose dual variables have the highest value (see figure 41). Applying theorem 5.2, we have the dual variables computed for the infinity norm on figure 42.

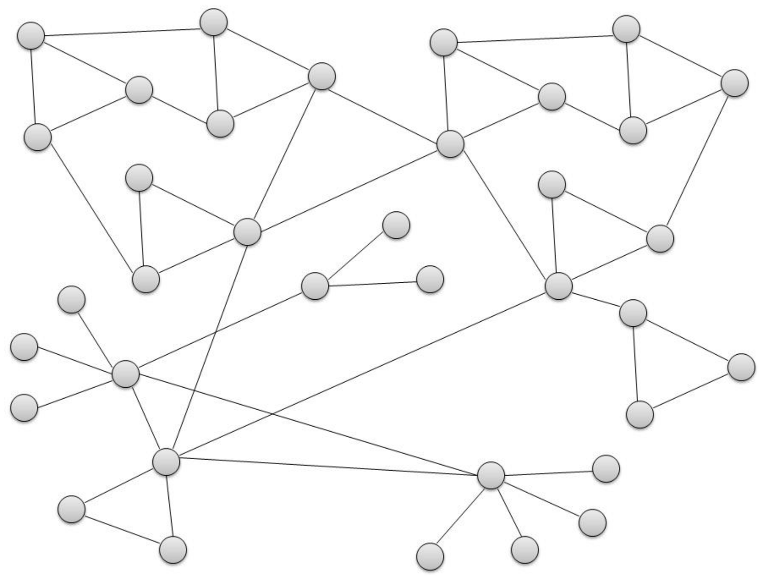

Figure 38: Example of a general graph.

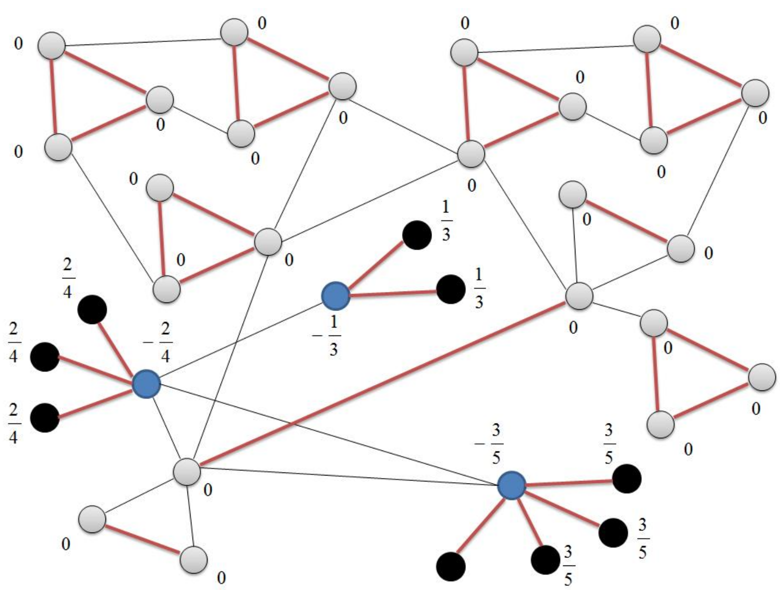

Figure 39: The NNLS solution.

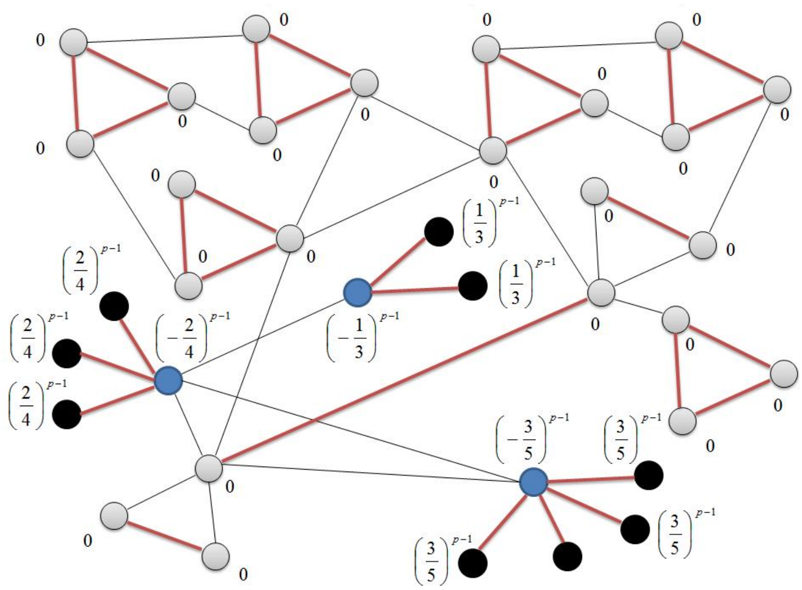

Figure 40: P-Norm solution, for $1 < p < \infty$.

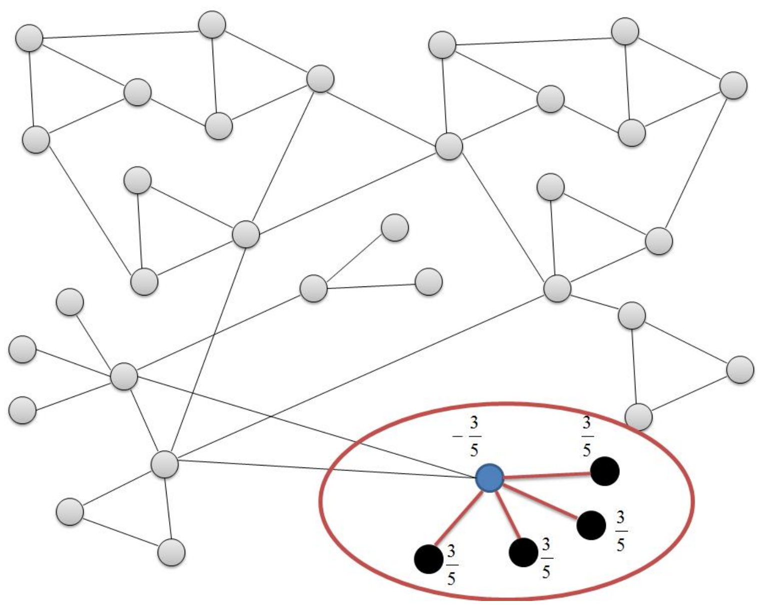

Figure 41: Connected component with maximum value of the dual variables.

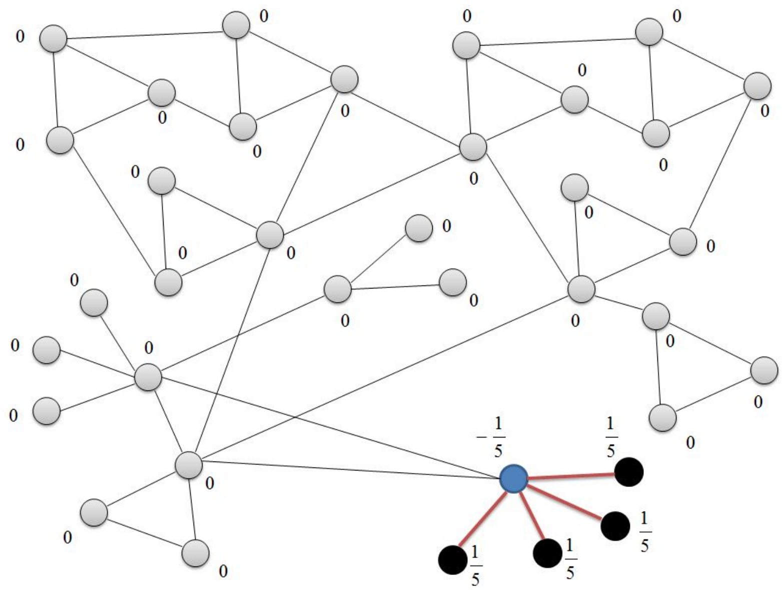

Figure 42: The infinity norm solution.

CHAPTER VI

THE ONE-STEP NNLS METHOD FOR THE ASSIGNMENT PROBLEM

In this chapter, we describe how we can solve the assignment problem by using the NNLS algorithm to solve a single norm 2 problem, as opposed to the NNLS primal-dual algorithm presented on chapter 2.

Throughout this section, n will denote the number of workers and also the number of jobs. It will be denoted by A the incidence matrix of the bipartite graph that represents the possible assignments of workers to jobs. We will denote the column of A that represents the assignment of worker i to job j by $A_{(i,j)}$.

6.1 The Modified Least Squares Problem

As we have seen before, the assignment problem can be modeled as the following LP:

$$
\begin{aligned}
(P): \min \quad & \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n} c_{(i,j)} x_{(i,j)} \\
\text{s.t.} \quad & \\
& Ax = b \\
& x \geq 0
\end{aligned}
$$

Now, let $r > 0$ and consider the following NNLS problem:

$$
\begin{aligned}
(Q_r) : \min \qquad & \textstyle\sum_{j=1}^{n} \frac{1}{2} s_j^2 + \frac{1}{2} t^2 \\
\text{s.t.} \qquad & \\
& Ax + s = b \qquad (35) \\
& \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n} \frac{c_{(i,j)}}{r} x_{(i,j)} + t = 0 \qquad (36) \\
& x \geq 0
\end{aligned}
$$

We will call r the normalization factor. Let x^p be the solution of problem (P) and x^r be the solution of problem (Q_r). A natural question arises: for which values of r (if any) is it possible to extract the solution x^p from x^r? In order to answer this, we will need to build up some theory.

First, let us write the KKT conditions for problem (Q_r). Thus, the vector (x, s, t) is a solution for (Q_r) if and only if there exists π and ρ such that:

$$
\begin{aligned}
(DQ_r) : \quad & \pi^t A_{(i,j)} + \tfrac{c_{(i,j)}}{r} \rho = 0 \quad \text{if } x_{(i,j)} > 0 \\
& \pi^t A_{(i,j)} + \tfrac{c_{(i,j)}}{r} \rho \leq 0 \\
& \pi_j = s_j \qquad \forall j \\
& \rho = t \qquad .
\end{aligned}
$$

Now, let z_p be the optimal solution value of (P) and z_r be the optimal solution value of (Q_r). From primal feasibility, since $x \geq 0$, $r > 0$ and $c_{(i,j)} \geq 0$, $\forall i, j$, we must have in the optimal that $\rho \leq 0$.

Moreover, from the KKT conditions and primal feasibility, it is easy to see that if $c > 0$, then the optimal solution of (Q_r) is never zero since:

$$\pi_k = 0 \ , \forall k \quad \Leftrightarrow \quad \exists x_{(i,j)} > 0 \quad \Leftrightarrow \quad \rho < 0.$$

Therefore, if A is an incidence matrix of a bipartite graph that has a perfect matching, for a fixed $r > 0$, we already know that in general we do not have $x^p = x^r$. Therein lies the need of some "rounding scheme" whereby some values of x^r are to be set to zero and the others to one, making it possible to extract the solution x^p from x^r.

Thus, we want to know for which values of $r > 0$ (if any), there can be a "rounding scheme" such that it is possible to extract the solution x^p from x^r.

In order to ensure that we are setting the correct value of x^r to zero/one, we must have that the optimal basis of the NNLS algorithm for problem (Q_r) contains the optimal one for the primal-dual NNLS applied to problem (P). Thus, since the optimal basis for the primal-dual NNLS applied to problem (P) will be a maximum matching, then we must find for which values of $r > 0$ the optimal basis of the NNLS algorithm for problem (Q_r) will also be a perfect matching or some graph from which the perfect matching can be easily extracted. It is important to recall that a perfect matching is a forest whose trees have at most one edge and without isolated nodes.

It is clear that a feasible basis for the NNLS algorithm for problem (Q_r) will be a forest (i.e., a set of trees). First, let us find for which values of $r > 0$ the NNLS algorithm for problem (Q_r) must end with an optimal basis that does not have any isolated nodes.

Suppose that A be an incidence matrix of a bipartite graph that is not a matching, i.e., there exists more than one possible matching for the underlying graph, otherwise it is trivial.

Lemma 25. *Let A be an incidence matrix of a bipartite graph the has a complete matching. If $r > \|c\|_1$ for problem (Q_r), then $\pi_i < 1$, $\forall i$.*

Proof. Suppose, for a contradiction, that the implication does not hold. Let (π, ρ) be the optimal values of the dual variables of problem Q_r, for $r \geq \|c\|_1 + 1$, with $\pi_k = 1$.

Thus, since $s = \pi^r$ and $\rho^r = t$:

$$z_r = \sum_{j=1}^{n} \frac{1}{2} s_j^2 + \frac{1}{2} t^2$$
$$z_r = \sum_{j=1}^{n} \frac{1}{2} (\pi^r)_j^2 + \frac{1}{2} (\rho^r)^2 \geq \frac{1}{2}$$

Since A is an incidence matrix of a bipartite graph the has a complete matching, let $\hat{x}$ be a primal feasible solution that is complete matching, i.e., $\hat{x}_i = 1$ for the vertices i that are in the matching, and $\hat{x}_i = 0$ otherwise. Let $(\hat{\pi}, \hat{\rho})$ be the value of the dual variables corresponding to this solution. Let $\hat{z}_r$ be the value of the objective function of this solution. Then:

$$\hat{z}_r = \sum_{j=1}^{n} \frac{1}{2} \hat{\pi}_j^2 + \frac{1}{2} \hat{\rho}^2$$

Since $\hat{x}$ is a complete matching, then $\hat{\pi}_i = 0$, $\forall i$.

$$\hat{z}_r = \sum_{j=1}^{n} \frac{1}{2} \hat{\rho}^2 \leq \frac{1}{2} \left(\frac{\|c\|_1}{\|c\|_1 + 1}\right)^2 < \frac{1}{2}$$

Combining the inequalities obtained:

$$\hat{z}_r < \frac{1}{2} \leq z_r$$
$$\hat{z}_r < z_r \Rightarrow z_r \quad \text{is not optimal!}$$

Therefore, if $r = \|c\|_1 + 1$ and (π, ρ) are the optimal values of the dual variables of problem Q_r, then we must have that $\pi_i < 1$, $\forall i$. □

Since lemma 25 implies that for any value of $r > \|c\|_1$ the optimal solution obtained by the NNLS algorithm will not contain an isolated node, it would suffice now to show that there exist a value of r such that each tree of the forest generated by the optimal basis of the NNLS algorithm will have at most one edge (since there can be not isolated nodes, each tree will have exactly one edge).

However, it is not possible to satisfy this requirement for every cost matrix. Consider the example shown in figure 43, where the costs are indicated on the arcs. No matter how much we increase r (the normalization factor), the NNLS solution will NOT be a set of trees each with one edge only. For r large enough, we will have the optimal basis shown on figure 44.

Therefore, we must relax the requirement that each tree of the forest generated by the optimal basis of the NNLS algorithm will have at most one edge. Thus, we will only require that every tree of the forest generated by the optimal basis of the NNLS algorithm will have a perfect matching. As we shall see later, it is very easy to extract a perfect matching from a tree.

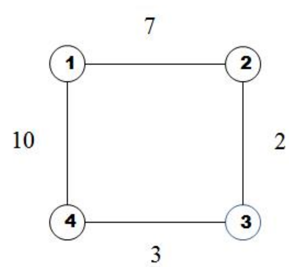

Figure 43: Assignment problem on 4 nodes, with cost $c = [7, 2, 3, 10]^t$

Recall from chapter 3 that every feasible basis for the NNLS algorithm applied to problem (Q_r) is a tree, if we do not take into account constraint 36. It is easy to see that if a subset of the columns of A is linearly independent, then it will still be linearly independent with the inclusion of the constraint 36. This means that every tree (more generally, a forest) is a basis for problem (Q_r).

In order to see that every connected component of any basis is a tree, let us take a look at the conditions that must be satisfied for a basis.

Let $\tilde{A} = [A^t \ c]^t$ and $\tilde{b} = [b^t \ 0]^t$, where $b_j = 1$, $\forall j$. Let $\tilde{B}_r$ be a feasible basis of

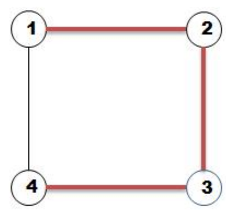

Figure 44: Optimal basis.

the NNLS for problem (Q_r), i.e., a linearly independent subset of the columns of $\tilde{A}$ and let $I_{\tilde{B}_r}$ and I_B be the index set of the columns in $\tilde{B}_r$ and B, respectively. In order to obtain the current primal and dual solutions we need to solve the following unrestricted least squares problem:

$$\min_x \|\tilde{B}_r x - \tilde{b}\| \tag{37}$$

Let $y^r = (x^r, \pi^r, \rho^r)$ be the primal-dual pair that is the solution of problem 37 for a fixed $r > 0$. Then, (x^r, π^r, ρ^r) must satisfy:

$$(\pi^r)^t A_{(i,j)} + \frac{c_{(i,j)}}{r}\rho^r \quad = 0 \tag{38}$$

$$\sum_{(i,j)\in I_B} x^r_{(i,j)}\frac{c_{(i,j)}}{r} + \rho^r \quad = 0 \tag{39}$$

$$Bx^r + \pi^r \quad = 1 \tag{40}$$

Lemma 26. *Let $\tilde{B}_r = [B^t \ c_B]^t$, where c_B is the vector that contains the components of c that correspond to the columns of B. If B is a forest then $\tilde{B}_r$ is a basis for the NNLS algorithm, i.e., 38-40 has unique solution that is given by $y^r = ((\tilde{B}_r)^t\tilde{B}_r)^{-1}(\tilde{B}_r)^t\tilde{b}$.*

Proof. This is clear, since B is a submatrix of $\tilde{B}_r$ whose set of columns is linearly independent. □

The converse of lemma 26 is not true. In order to check that, consider the following graph (a cycle with 4 edges) shown on figure 45. The number on the edges are their corresponding coefficients in the objective function, i.e., $c = [5, 3, 1, 4]^t$. The unrestricted LS solution computed by the pseudoinverse is shown on figure 46 with the value on each edge being the primal solution. After with change the cost of the edge $(3, 4)$ from 1 to 2 (see figure 47) we have that the new graph cannot have its only cycle as a basis, since:

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 5 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 3 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 2 \end{pmatrix}.$$

Therefore, different from the NNLS algorithm for bipartite graphs studied on chapter 4, in order to check whether or not a subgraph is a valid basis for the NNLS algorithm, we must take into account both the topology and the cost vector (objective function) of the subgraph. However, as the next lemma states, if we start with a feasible basis that is a forest, the NNLS algorithm will terminate with a basis that is also a forest.

Lemma 27. *Suppose that the initial basis used in the NNLS algorithm is a forest, then, at any iteration, the basis $\tilde{B}_r$ will NOT contain the edge set of a cycle.*

Proof. Suppose, for a contradiction, that, at some iteration, the basis $\tilde{B}_r$ contains the edge set of a cycle $C = \{(1, 2), (2, 3), ..., (2k - 1, 2k), (2k, 1)\}$, for some integer k. Let $I_C = \{1, ..., 2k\}$. Let us denote O and E the set of odd and even natural numbers, respectively. Let (π^r, ρ^r) be the dual solution corresponding to the basis $\tilde{B}_r$. Dual feasibility implies that:

$$\pi_i^r + \pi_{i+1}^r + \frac{c_{(i,i+1)}}{r}\rho^r = 0 \quad i \in I_C \cap O \tag{41}$$

$$\pi_i^r + \pi_{i+1}^r + \frac{c_{(i,i+1)}}{r}\rho^r = 0 \quad i \in I_C \cap E \tag{42}$$

$$\pi_{2k}^r + \pi_1^r + \frac{c_{(2k,1)}}{r}\rho^r = 0 \tag{43}$$

Multiplying equalities 42 and 43 by -1 and adding to the sum of equalities 41, we have:

$$\begin{aligned}\sum_{(u,v)\in C|u\in I_C\cap O}\frac{c_{(u,v)}}{r}\rho^r - \sum_{(u,v)\in C|u\in I_C\cap E}\frac{c_{(u,v)}}{r}\rho^r &= 0\\ \Big(\sum_{(u,v)\in C|u\in I_C\cap O} c_{(u,v)} - \sum_{(u,v)\in C|u\in I_C\cap E} c_{(u,v)}\Big)\frac{\rho^r}{r} &= 0\end{aligned}$$

Since $\rho^r \neq 0$:

$$\sum_{(u,v)\in C|u\in I_C\cap O} c_{(u,v)} - \sum_{(u,v)\in C|u\in I_C\cap E} c_{(u,v)} = 0 \tag{44}$$

Since we assumed that the initial basis did not contain a cycle, this means that in a previous iteration not all of the edges in C were in the basis. Suppose, without loss of generality, that the last edge of C of enter the basis is $(2k,1)$. Let $(\hat{\pi}^r, \hat{\rho}^r)$ be the dual solution corresponding to this basis. Dual feasibility implies that:

$$\hat{\pi}_i^r + \hat{\pi}_{i+1}^r + \frac{c_{(i,i+1)}}{r}\hat{\rho}^r = 0 \quad i \in I_C \cap O \tag{45}$$

$$\hat{\pi}_i^r + \hat{\pi}_{i+1}^r + \frac{c_{(i,i+1)}}{r}\hat{\rho}^r = 0 \quad i \in I_C \cap E \tag{46}$$

Since the edge $(2k,1)$ wants to enter the basis:

$$\hat{\pi}_{2k}^r + \hat{\pi}_1^r + \frac{c_{(2k,1)}}{r}\hat{\rho}^r > 0 \tag{47}$$

Proceeding as before, i.e., multiplying equalities 46 and inequality 47 by -1 and adding to the sum of equalities 45, we have:

$$\begin{aligned}\sum_{(u,v)\in C|u\in I_C\cap O}\frac{c_{(u,v)}}{r}\hat{\rho}^r - \sum_{(u,v)\in C|u\in I_C\cap E}\frac{c_{(u,v)}}{r}\hat{\rho}^r &> 0\\ \Big(\sum_{(u,v)\in C|u\in I_C\cap O} c_{(u,v)} - \sum_{(u,v)\in C|u\in I_C\cap E} c_{(u,v)}\Big)\frac{\hat{\rho}^r}{r} &> 0\end{aligned}$$

Since $\rho^r < 0$ (primal feasibility) and $r > 0$:

$$\sum_{(u,v)\in C|u\in I_C\cap O} c_{(u,v)} - \sum_{(u,v)\in C|u\in I_C\cap E} c_{(u,v)} < 0$$

The last inequality obtained contradicts equality 44. $\square$

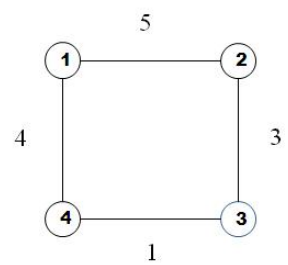

Figure 45: Cycle with cost vector $c = [5, 3, 1, 4]^t$

Let (x^r, π^r, ρ^r) be the solution of the problem (Q_r).

Let F^r be the forest induced by the optimal basis $\tilde{B}_r$. Let us define the following sets:

$$\Omega_r^+ = \{v | v \in F^r, \pi_v^r > 0\} \tag{48}$$

$$\Omega_r^- = \{v | v \in F^r, \pi_v^r < 0\} \tag{49}$$

$$\Omega_r^0 = \{v | v \in F^r, \pi_v^r = 0\} \tag{50}$$

Lemma 28. *Let F^r be defined as previously and $\tilde{B}_r = [B^t \ c_B]^t$. Let H^r be a connected component of F^r. Let π^r be defined as before. Then, we have:*

$$\forall k \in \Omega_r^- \cap V(H^r) \quad \exists j \in \Omega_r^+ \cap V(H^r) \text{ such that } \quad |\pi_j^r| \geq |\pi_k^r| \ .$$

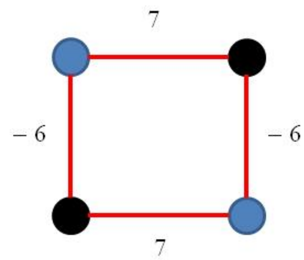

Figure 46: Valid basis (not feasible) for the NNLS algorithm.

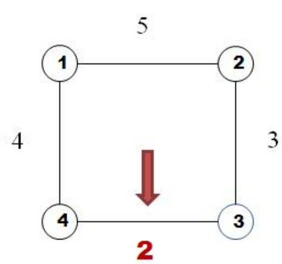

Figure 47: Cycle with cost vector changed.

Proof. if $|V(H^r)| = 1$, then the lemma is trivially true.

Let $v \in \Omega_r^- \cap V(H^r)$. Since $|V(H^r)| > 1$, $\exists u \in V(H^r)$ such that $(v,u) \in H^r$. From the optimality conditions for (Q_r) we have:

$$\pi_v^r + \pi_u^r + \frac{c_{(v,w)}}{r}\rho = 0$$

Since $\rho \leq 0$ and $\frac{c_{(v,w)}}{r} \geq 0$:

$$\pi_v^r + \pi_u^r \geq 0$$

$$\pi_u^r \geq -\pi_v^r = |\pi_v^r|$$

$$\Rightarrow \pi_u^r \geq |\pi_v^r|$$

□

Theorem 29. *Let F^r be defined as previously and $\tilde{B}_r = [B^t \; c_B]^t$. Let H^r be a connected component of F^r that does not have a perfect matching. Let π^r be defined as before. Then, we have:*

$$\exists k \in V(H^r) \qquad \textit{such that} \qquad |\pi_k^r| \geq \frac{1}{2n\sqrt{2n}} \; .$$

Proof. Suppose for a contradiction that no such k exists. If $\tilde{B}_r$ is a basis, then let (x^r, s^r, t^r) are the solution of the following NNLS problem:

$$
\begin{aligned}
(Q_r) : \min \qquad & \textstyle\sum_{j=1}^n \frac{1}{2}s_j^2 + \frac{1}{2}t^2 \\
\text{s.t.} \qquad & \\
& Ax + s = b \\
& \textstyle\sum_{i=1}^n \sum_{j=1}^n \frac{c_{(i,j)}}{r} x_{(i,j)} + t = 0 \\
& x \geq 0
\end{aligned}
$$

Let us define the following quantity:

$$t^* = -\sum_{(i,j) \in F^r - H^r} c_{(i,j)} x_{(i,j)}$$

Let A_H be the submatrix of A that contains the columns corresponding to the edges of the connected component H^r.

Consider now the following NNLS problem:

$$
\begin{array}{lc}
(Q_r^H) : \min & \sum_{j \in V(H)} \frac{1}{2}\eta_j^2 \\
\text{s.t.} & \\
 & A_H y + \eta = b \\
 & \sum_{(i,j)\in H^r} \frac{c_{(i,j)}}{r} y_{(i,j)} + (t^r - t^*) = 0 \\
 & x \geq 0
\end{array}
$$

Let s_H^r be the subvector of s^r whose elements correspond to the component H.

Let (y^r, η^r) be the optimal solution of (Q_r^H).

Claim 30. $\eta^r = s_H^r$.

Proof. Let $\hat{z}_H$ be the solution value of problem (Q_r^H). Let us define

$$
z_H^r = \sum_{i \in V(H)} \frac{1}{2}(s_j^r)^2
$$

Suppose $\eta^r \neq s_H^r$. Since the objective function of both problems is strictly convex, their solution is unique (on s and z). Thus, since $\eta^r \neq s_H^r$, we must have either $\hat{z}_H < z_H^r$ or $\hat{z}_H > z_H^r$.

If $\hat{z}_H > z_H^r$, then η^r cannot be optimal, since s_H^r is feasible for (Q_r^H) and its cost in the objective function is smaller.

If $\hat{z}_H < z_H^r$, then s^r cannot be optimal, since we can build a primal feasible solution for (Q_r) $(\overline{x}, \overline{s}, \overline{t})$ by setting

$$
\begin{aligned}
\overline{x}_{(i,j)} &= x_{(i,j)}^r \ , \ (i,j) \in F - H \\
\overline{x}_{(i,j)} &= y_{(i,j)}^r \ , \ (i,j) \in H \\
\overline{s}_i &= s_i^r \ , \ i \in V(F) - V(H) \\
\overline{s}_i &= \eta_i^r \ , \ i \in V(H)
\end{aligned}
$$

with cost in the objective function that is smaller. $\square$

Consider now the problem (P^H):

$$(P^H) : \min \quad \textstyle\sum_{j \in H} \frac{1}{2}\eta_j^2$$
$$\text{s.t.}$$
$$A_H y + \eta = b$$
$$x \geq 0$$

Let z_H^p be the optimal value of the objective function of (P^H). Since (P^H) is a relaxation of (Q_r^H), then $z_H^p \leq \hat{z}_H$. Problem (P^H) is NNLS problem whose constraint matrix is the incidence matrix of a bipartite graph. Let $(y^*, \eta*)$ be the optimal solution of P^H. In chapter 4 we know that the edges (i, j) corresponding to the components of y^* such that $y^*_{(i,j)} > 0$ form a forest $F^* \subset F^r$. Since H does not have a perfect matching, at least one of these connected components has a vertex whose dual variable is not zero. Let $H^* \subset F^*$ be this connected component. Let Δ and Ξ be the partitions of H^*. Recall from chapter 3 we have the following formula for the dual variables corresponding to vertices in H^*:

$$\eta_j^* = \frac{|\Delta| - |\Xi|}{|\Delta| + |\Xi|} \ , \ j \in \Delta$$
$$\eta_j^* = \frac{|\Xi| - |\Delta|}{|\Delta| + |\Xi|} \ , \ j \in \Xi$$

Since $\eta_j^* \neq 0$ and $|\Delta|$ and $|\Xi|$ are integers, then $|\Delta| - |\Xi| \geq 1$.
Since $|\Delta| + |\Xi| \leq |H^*| \leq |F^*| \leq |F^r| \leq |V(G)| \leq 2n$, then

$$\eta_j^* = \frac{|\Delta| - |\Xi|}{|\Delta| + |\Xi|} \geq \frac{1}{|\Delta| + |\Xi|} \geq \frac{1}{2n}$$

Thus, we have the following lower bound:

$$\hat{z}_H \geq z_H^r \geq z_H^p \geq \sum_{j \in V(F^*)} \eta_j^* \geq \sum_{j \in V(H^*)} \eta_j^* \geq (\frac{1}{2n})^2$$
$$\Rightarrow \hat{z}_H \geq \frac{1}{4n^2} \tag{51}$$

Using our assumption that $|\pi_k^r| < \frac{1}{2n\sqrt{2n}}$:

$$\hat{z}_H \leq \sum_{j \in V(H^r)} \pi_j^2 < |V(H^r)|(\frac{1}{2n\sqrt{2n}})^2 \leq 2n(\frac{1}{2n\sqrt{2n}})^2$$

$$\Rightarrow \hat{z}_H < \frac{1}{4n^2} \tag{52}$$

Clearly, relation 52 contradicts 51.

Thus, for at least for one $k \in V(H^r)$ we must have that $|\pi_k^r| \geq \frac{1}{2n\sqrt{2n}}$. □

Corollary 31. *Let F^r be defined as previously and $\tilde{B}_r = [B^t \ c_B]^t$. Let H^r be a connected component of F^r that does not have a perfect matching. Let π^r be defined as before. Then, we have:*

$$\exists k \in \Pi_r^+ \cap V(H^r) \quad \textit{such that} \quad \pi_k^r \geq \frac{1}{2n\sqrt{2n}} \ .$$

Proof. Straightforward application of theorem 29 and lemma 28. □

The following lemma provides a comparison between the cardinality of the sets Ω_r^- and Ω_r^+ for each tree in F^r.

Lemma 32. *If $v \in \Omega_r^-$ and $(v, u) \in T^r$, then $u \in \Omega_r^+$. Moreover, $|T^r \cap \Omega_r^-| < |T^r \cap \Omega_r^+|$.*

Proof. Suppose for a contradiction, that there exists two nodes v, w such that $(v, w) \in T^r$ and $v, w \in \Omega_r^-$. From the optimality conditions for (Q_r) we have:

$$\pi_v^r + \pi_w^r + \frac{c_{(v,w)}}{r}\rho = 0 \tag{53}$$

Since $x \geq 0$ (primal feasibility) and $c \geq 0$, then $\rho \leq 0$. If $v, w \in \Omega_r^-$ then

$$\pi_v^r + \pi_w^r + \frac{c_{(v,w)}}{r}\rho < \frac{c_{(v,w)}}{r}\rho \leq 0$$

contradicting 53.

To prove the second assertion, it suffices to observe that any leaf $u \in T^r$ has to belong to Ω_r^+ (otherwise, primal feasibility implies that $\exists j \in T^r$ such that $x_{ju} > 1$). Therefore $|T^r \cap \Omega_r^-| < |T^r \cap \Omega_r^+|$. $\square$

If the optimal forest contains more than one connected component (i.e., more than one tree), then following lemma proves the existence of an arc connecting one vertex in Ω_r^+ to another in Ω_r^0.

Lemma 33. *Let F^r be defined as previously and let $u \in F^r \cap \Omega_r^+$ be chosen as in corollary 31. Then, there exists $v \in (\Omega_r^+ \cup \Omega_r^0)$ such that $(u, v) \in E(G)$.*

Proof. Suppose, for a contradiction, that such edge does not exist. Since the graph G is a complete bipartite graph, then if $(u, v) \notin E(G)$, u and v belong to the same partition of G. Let us denote the partition of G by Δ and Ξ. Suppose, without loss of generality, that $u, v \in \Delta$. Following the same reasoning:

$$\Omega_r^0 \cup \Omega_r^+ \subset \Delta \quad \Rightarrow \quad \Xi \subset \Omega_r^-$$

From lemma 32, we have that $|\Omega_r^+| > |\Omega_r^-|$, then:

$$|\Delta| \geq |\Omega_r^0 \cup \Omega_r^+| \geq |\Omega_r^0| \cup |\Omega_r^+| > |\Omega_r^-| \geq |\Xi|$$
$$\Rightarrow |\Delta| > |\Xi|$$

Contradiction, since $|\Delta| = |\Xi|$. $\square$

The following lemma provides an upper bound on the value of $|\rho^r|$:

Lemma 34. *Let (π^r, ρ^r) the dual solution for the problem Q_r. Then $|\rho^r| \leq \sqrt{2n}$.*

Proof. Let $(\hat{s}, \hat{t})$ be the optimal solution of problem Q_r with value z_r.
Since $x = 0$ and $s_j = 1$, $\forall j$ is a feasible primal solution, then:

$$\sum_{j=1}^{n} \hat{s}_j^2 + \hat{t}_j^2 = z_r \leq 2n$$

Thus:

$$|\rho^r| = |\hat{t}| = \sqrt{\hat{t}^2} \leq \sqrt{\sum_{j=1}^{n} \hat{s}_j^2 + \hat{t}_j^2} \leq \sqrt{2n}$$

□

Now we are ready to prove the main result of this section: how large the value of r must be in order to allow us to extract the norm 1 solution from problem (Q_r).

Theorem 35. *Let A be an incidence matrix of a bipartite graph the has a complete matching. Suppose that $c \neq 0$. If $r > 4n^2\|c\|_\infty$, then each tree of the forest generated by the optimal basis of the NNLS algorithm for the problem (Q_r) will have a perfect matching.*

Proof. If $\tilde{B}_r$ is an optimal basis for (Q_r) and $(v, w) \in E(G)$ is not in the basis, then we have:

$$\pi_v^r + \pi_w^r + \frac{c_{(v,w)}}{r}\rho^r \leq 0 \tag{54}$$

Thus, if we want the value of r such that every tree in the optimal basis has a perfect matching, it suffices to check for which value of r the inequality 54 will not be satisfied for some edge (v, w), if the tree does not have a perfect matching. Thus, if a feasible basis induces a tree that does not have a perfect matching, then $\Omega_r^+ \neq \emptyset$ (see corollary 31). Then, from lemma 33, there exists must exist $v \in \Omega_r^+$ and $w \in \Omega_r^+ \cup \Omega_r^0$ such that $(v, w) \in E(G)$. Thus:

$$\begin{aligned} \pi_v^r + \pi_w^r + \frac{c_{(v,w)}}{r}\rho^r &> 0 \\ \pi_v^r + \pi_w^r &> \frac{c_{(v,w)}}{r}|\rho^r| \\ r &> \frac{c_{(v,w)}}{\pi_v^r + \pi_w^r}|\rho^r| \end{aligned}$$

Suppose, without loss of generality, that vertex v is chosen in order to satisfy corollary

31. Thus

$$r > \frac{c_{(v,w)}}{\frac{1}{2n\sqrt{2n}} + 0}|\rho^r|$$
$$r > 2n\sqrt{2n}c_{(v,w)}|\rho^r| \tag{55}$$

Since we want inequality 55 to hold for every edge (v, w):

$$r > 2n\sqrt{2n}\|c\|_\infty|\rho^r|$$

Using lemma 27:

$$r > 2n\sqrt{2n}\|c\|_\infty\sqrt{2n} = 4n^2\|c\|_\infty$$

□

From lemma 25, if G is a bipartite graph with a perfect matching, then if $r > \|c\|_1$ then the solution of problem (Q_r) will not contain isolated nodes. From theorem 35, if $r > 4n^2\|c\|_\infty$, then every tree with more than one vertex in the solution will have a perfect matching. Since $4n^2\|c\|_\infty > \|c\|_1$, if we choose $r > 4n^2\|c\|_\infty$, then our solution for (Q_r) is guaranteed to be a forest with a perfect matching. It suffices to show that this perfect matching is indeed of minimum cost.

Theorem 36. *Let $r > 4n^2\|c\|_\infty$ and let $\tilde{B}_r$ be an optimal basis for (Q_r). Then the perfect matching contained in the forest is the minimum cost complete matching.*

Proof. Let F^r be the edge set of the columns in the basis. By the choice of r, from theorem 35, F^r contains a perfect matching. Let M^r be is this complete matching. We have to show that it is of minimum cost. Suppose for a contradiction that there exists another complete matching M with cost less that M^r. Thus, there exists an even cycle $C = \{(1, 2), (2, 3), ..., (2k - 1, 2k), (2k, 1)\}$, for some integer k such that:

$$\sum_{(u,v)\in M\cap C} c_{(u,v)} < \sum_{(u,v)\in M^r\cap C} c_{(u,v)} \tag{56}$$

Let $I_C = \{1, ..., 2k\}$. Let us denote O and E the set of odd and even natural numbers, respectively. Without loss of generality, let

$$\begin{aligned} M^r &= \{(i, i+1) \mid i \in I_C \cap O\} \\ M &= \{(i, i+1) \mid i \in I_C \cap E\} \end{aligned}$$

Thus:

$$\sum_{i \in I_C \cap O} c_{(i,i+1)} > \sum_{i \in I_C \cap E} c_{(i,i+1)} \tag{57}$$

Since M^r has the columns of the optimal basis:

$$\pi_v^r + \pi_w^r + \frac{c_{(v,w)}}{r}\rho^r = 0 \quad \forall (v,w) \in M^r \cap C \tag{58}$$

$$\pi_v^r + \pi_w^r + \frac{c_{(v,w)}}{r}\rho^r \leq 0 \quad \forall (v,w) \in M \cap C \tag{59}$$

Summing up all equalities 58 and multiplying the result 58 by -1 and adding this value to the sum of all inequalities in 59, we have:

$$\sum_{i \in I_C \cap O} c_{(i,i+1)} - \sum_{i \in I_C \cap E} c_{(i,i+1)} \leq 0$$

$$\sum_{i \in I_C \cap O} c_{(i,i+1)} \leq \sum_{i \in I_C \cap E} c_{(i,i+1)} \tag{60}$$

Inequality 60 contradicts our assumption in 57, thereby proving the theorem. □

Theorems 35 and 36 provide us with a way of selecting r such that if we apply the NNLS algorithm to problem (Q_r) starting from a feasible basis that is a forest, then the optimal basis will be a forest that has a perfect matching.

At this point, one question arises: how to extract the perfect matching from this forest? The following lemma provides a simple way of doing that.

Lemma 37. *A perfect matching can be extracted from a tree with m edges in m steps.*

Proof. We will use induction on the number of edges of the tree. For a tree with one edge, the theorem is trivially true.

Suppose the theorem is valid for any tree with number of edges less than m.

Let T be a tree with m edges.

Since T is a tree, $\exists v \in T$ such that v is a leaf. Suppose that $(v, w) \in T$. The perfect matching on T will be (v, w) plus the perfect matching on $T - \{v, w\}$ (using induction hypothesis). □

6.2 Avoiding The Computation of The Normalization Factor

As we have seen on the previous section, we can compute the value of r of problem (Q_r) (defined in section 6.1) in such a way that if we apply the NNLS algorithm to the problem (Q_r), the optimal basis will be a forest that contains the perfect matching of minimum cost. In this section it will be shown that, in fact, we do not need to compute this normalization factor, leading to a method that is numerically stable even if the product $4n^2\|c\|_\infty$ is very large.

Therefore, our objective is to build an algorithm that just assumes $r > 0$, and arbitrarily large, and finds the optimal perfect matching.

For an edge (u, v), let us define

$$\bar{c}_{(u,v)} = \pi_u^r + \pi_v^r + \frac{c_{(u,v)}}{r}\rho \tag{61}$$

In terms of the new notation, it is clear that (π^r, ρ^r) is an optimal dual solution for (Q_r) if and only if there exists an x primal feasible for (Q_r) such that:

$$\begin{array}{rcl} \bar{c}_{(u,v)} & = & 0 \quad \text{if } x_{(u,v)} > 0 \\ \bar{c}_{(u,v)} & \leq & 0 \quad \forall (u, v) \in G \end{array}$$

Let T be a tree and define $v \in T$ its root. Given a root If $w \in T - \{v\}$, let P_w be the unique path that connects w to v in T and let $E(P_{vw})$ be the set of edges in this path. Let $I_{P_{vw}} = \{1, ..., |P_w|\}$, and O, E be the set of odd and even numbers, respectively. For any vertex $w \in T$, let us label the vertices in this unique path

$v_0 = v, \ldots, v_{|E(P_{vw})|} = w$. Let us define the following sets:

$$\begin{aligned} P^O_{vw} &= \{(v_{j-1}, v_j) | j \in I_{P_{vw}} \cap O\} \\ P^E_{vw} &= \{(v_{j-1}, v_j) | j \in I_{P_{vw}} \cap E\} \end{aligned}$$

Lemma 38. *Let T be a connected component of a basis of the NNLS algorithm. Let v be a leaf of T. Then if w is any vertex of T, we have that*

$$\pi_w = (-1)^{|E(P_w)|}\pi_v + (\sum_{(s,t)\in P^E_{vw}} c_{(s,t)} - \sum_{(s,t)\in P^O_{vw}} c_{(s,t)})\frac{\rho}{r} .$$

Proof. We will prove by induction on the number of edges of the tree T.

Suppose T has only one edge. Let $(w, v) \in T$. Then we have $|E(P_w)| = 1$.

$$\begin{aligned} \pi^r_w + \pi^r_v + \frac{c_{(w,v)}}{r}\rho = 0 \\ \pi^r_w = -\pi^r_v - \frac{c_{(w,v)}}{r}\rho \\ \pi^r_w = (-1)^1\pi^r_v + (-c_{(w,v)})\frac{\rho}{r} \end{aligned}$$

Then the formula is true, since $|E(P_{vw})| = 1$, $P^E_{vw} = \emptyset$ and $P^O_{vw} = \{(w, v)\}$.
Let T be a tree with m edges. Suppose the theorem is true for every tree with less than m edges.

Let (w, u) be an arbitrary edge of T. Suppose, without loss of generality, that u is not a leaf. Let $v \in T - \{w\}$ be a leaf that is connected to u in $T - \{w\}$. Since $T - \{w\}$ has less than m edges, using induction hypothesis

$$\pi^r_u = (-1)^{|E(P_u)|}\pi^r_v + (\sum_{(s,t)\in P^E_{vu}} c_{(s,t)} - \sum_{(s,t)\in P^O_{vu}} c_{(s,t)})\frac{\rho}{r} \tag{62}$$

The optimality condition for the arc (w, u) is:

$$\pi^r_w + \pi^r_u + \frac{c_{(w,u)}}{r}\rho = 0 \tag{63}$$

Substituting 62 in 63:

$$\pi_w^r + (-1)^{|E(P_{vu})|}\pi_v^r + (\sum_{(s,t)\in P_{vu}^E} c_{(s,t)} - \sum_{(s,t)\in P_{vu}^O} c_{(s,t)})\frac{\rho}{r} + \frac{c_{(w,u)}}{r}\rho = 0$$
$$\pi_w^r = -(-1)^{|E(P_{vu})|}\pi_v^r - (\sum_{(s,t)\in P_{vu}^E} c_{(s,t)} - \sum_{(s,t)\in P_{vu}^O} c_{(s,t)})\frac{\rho}{r} - c_{(w,u)}\frac{\rho}{r}$$
$$\pi_w^r = (-1)^{|E(P_{vu})+1|}\pi_v^r + (\sum_{(s,t)\in P_{vu}^O} c_{(s,t)} - \sum_{(s,t)\in P_{vu}^E\cup\{(u,w)\}} c_{(s,t)})\frac{\rho}{r}$$

Thus, the formula is true, since:

$$\begin{aligned} |E(P_{vw})| &= |E(P_{vu})| + 1 \\ P_{vw}^E &= P_{vu}^O \\ P_{vw}^O &= P_{vu}^E \cup \{(w,v)\} \end{aligned}$$

□

Let T be a connected component of a basis of the NNLS algorithm. Let Δ and Ξ be the partitions of T. Let

$$\begin{aligned} \Delta^* &= \textstyle\sum_{j\in\Delta} b_j \\ \Xi^* &= \textstyle\sum_{j\in\Xi} b_j \end{aligned}$$

Then we have the following lemma:

Lemma 39. *Let T be a connected component of a basis of the NNLS algorithm and let (π^*, ρ) be the dual solution of Q_r. Then*

$$\sum_{j\in\Delta} \pi_j^* - \Delta^* = \sum_{j\in\Xi} \pi_j^* - \Xi^*$$

Proof. Let us consider the constraints of problem (Q_r) that are contained in the incidence matrix of T:

$$\sum_{i\in\delta_k} x_i + \pi_k^* = b_k \quad k \in \Delta \tag{64}$$
$$\sum_{i\in\delta_k} x_i + \pi_k^* = b_k \quad k \in \Xi \tag{65}$$

Multiplying the equalities in 65 by -1 and adding to the constraints in 64:

$$\begin{aligned}\sum_{k\in\Delta}\pi_k^* - \sum_{k\in\Xi}\pi_k^* &= \sum_{k\in\Delta} b_k - \sum_{k\in\Xi} b_k\\ \sum_{k\in\Delta}\pi_k^* - \Delta^* &= \sum_{k\in\Xi}\pi_k^* - \Xi^*\end{aligned}$$

proving the lemma. $\square$

Let T be a connected component of a basis of the NNLS algorithm. Let $v \in T$ be a leaf of T. If $w \in T - \{v\}$, then let us define the following quantity:

$$\psi_w^u = \sum_{(s,t)\in P_{vw}^E} c_{(s,t)} - \sum_{(s,t)\in P_{vw}^O} c_{(s,t)} \ .$$

The following corollary enables us to compute the value of the chosen root as a function of ρ.

Corollary 40. *Let T be a connected component of a basis of the NNLS algorithm. Let v be a leaf of T that is chosen as the root. Let Δ and Ξ be the partitions of T, with $v \in \Delta$. Then if w is any vertex of T, we have that*

$$\pi_v = \frac{\sum_{w\in T\cap P_{vw}^O}\psi_w^u - \sum_{w\in T\cap P_{vw}^E}\psi_w^u}{|T|}\frac{\rho}{r} + \frac{\Delta^* - \Xi^*}{|T|} \ .$$

Proof. Substitute the formula obtained in lemma 38 in the equation of lemma 39 to reach the desired result. $\square$

Before the next corollary, for the sake of simplicity, let us define the following quantity, given that v is a leaf of T that is chosen as the root and $w \in T - \{v\}$:

$$\eta_v \ = \sum_{w\in T\cap P_{vw}^O}\psi_w - \sum_{w\in T\cap P_{vw}^E}\psi_w \ .$$

The next corollary provides us a way of computing the dual variable for any vertex other than the root.

Corollary 41. *Let T be a connected component of a basis of the NNLS algorithm. Let v be a leaf of T that is chosen as the root. Let Δ and Ξ be the partitions of T, with $v \in$ Delta. Then if w is any vertex of T, we have that*

$$\pi_w = ((-1)^{|E(P_v)|}\frac{\eta_v}{|T|} + \psi^v_w)\frac{\rho}{r} + \frac{\Delta^* - \Xi^*}{|T|} \ .$$

Proof. Substitute the formula obtained in lemma 40 in the equation of lemma 38 to reach the desired result. □

Lemma 42. *Let T be a connected component of a basis of the NNLS algorithm. Let v be a leaf of T that is chosen as the root. Then, if (u, w) is edge of T, we have that*

$$x_{u,w} = \zeta^v_{uw} + \gamma^v_{uw}\frac{\rho}{r}$$

where ζ^v_{uw} and γ^v_{uw} are rationals.

Proof. We will prove by induction on the number of edges of the tree T.

Suppose T has only one edge. Let $(w, v) \in T$.

$$\begin{aligned} x_{(w,v)} + \pi_v &= 1 \\ x_{(w,v)} &= 1 - \pi_v \\ x_{(w,v)} &= 1 + (-c_{(w,v)})\frac{\rho}{r} \\ &\quad \text{where } \zeta^v_{wv} = 1 \text{ and } \gamma^v_{wv} = -c_{(w,v)} \ . \end{aligned}$$

Let T be a tree with m edges. Suppose the theorem is true for every tree with less than m edges.

Let v be a leaf of T. Using corollary 41, for every vertex $u \in T$ we calculated the following quantity:

$$\pi_u = ((-1)^{|E(P_v)|}\frac{\eta_v}{|T|} + \psi^v_w)\frac{\rho}{r} + \frac{\Delta^* - \Xi^*}{|T|} \ . \tag{66}$$

Let (w,u) be an arbitrary edge of T. Suppose, without loss of generality, that u is not a leaf. Let $v \in T-\{w\}$ be a leaf that is connected to u in $T-\{w\}$. Since $T-\{w\}$ has less than m edges, using induction hypothesis we have for every edge $(u_j,u) \in T-\{w\}$, there exists rationals $\zeta^v_{u_ju}$ and $\gamma^v_{u_ju}$ such that:

$$x_{(u_j,u)} = \zeta^v_{u_ju} + \gamma^v_{u_ju}\frac{\rho}{r} \tag{67}$$

The primal constraint at vertex u is:

$$\sum_{u_j \in \delta_u - \{w\}} x_{(u_j,u)} + x_{(w,u)} + \pi_u = 1 \tag{68}$$

Substituting 66 and 67 in 68:

$$\begin{aligned}
\sum_{u_j \in \delta_u - \{w\}} (\zeta^v_{u_ju} + \gamma^v_{u_ju}\frac{\rho}{r}) + x_{(w,u)} + ((-1)^{|E(P_v)|}(\frac{\eta_v}{|T|} + \psi^v_w)\frac{\rho}{r} + \frac{\Delta^* - \Xi^*}{|T|} &= 1 \\
x_{(w,u)} = 1 - \sum_{u_j \in \delta_u - \{w\}} (\zeta^v_{u_ju} + \gamma^v_{u_ju}\frac{\rho}{r}) - ((-1)^{|E(P_v)|}(\frac{\eta_v}{|T|}) + \psi^v_w)\frac{\rho}{r} &- \frac{\Delta^* - \Xi^*}{|T|} \\
x_{(w,u)} = 1 - \sum_{u_j \in \delta_u - \{w\}} \zeta^v_{u_ju} - \frac{\Delta^* - \Xi^*}{|T|} + (-\sum_{u_j \in \delta_u - \{w\}} \gamma^v_{u_ju} + ((-1)^{|E(P_v)|+1}(\frac{\eta_v}{|T|}) - \psi^v_w))\frac{\rho}{r}
\end{aligned}$$

If we let

$$\begin{aligned}
\zeta^v_{wu} &= 1 - \sum_{u_j \in \delta_u - \{w\}} \zeta^v_{u_ju} - \frac{\Delta^* - \Xi^*}{|T|} \\
\gamma^v_{wu} &= -\sum_{u_j \in \delta_u - \{w\}} \gamma^v_{u_ju} + ((-1)^{|E(P_v)|+1}(\frac{\eta_v}{|T|}) - \psi^v_w)\ ,
\end{aligned}$$

since the rationality of ζ^v_{wu} and γ^v_{wu} can be trivially seen, the lemma is proved. $\square$

Lemma 43. *Let T be a connected component of a basis of the NNLS algorithm. Let v be a leaf of T that is chosen as the root. Then, if (u,w) is edge of T, we have that*

$$\rho = -(\frac{\sum_{(u,w)\in T} c_{(u,w)}\zeta^v_{uw}}{r^2 + \sum_{(u,w)\in T} c_{(u,w)}\gamma^v_{uw}})r$$

where the values of ζ^v_{uw} and γ^v_{uw} were obtained as in lemma 42.

Proof. Let us fix a leaf $v \in T$ to be the root.

The primal constraint that involves ρ is:

$$\sum_{(u,w)\in T} \frac{c_{(u,w)}}{r} x_{(u,w)} + \rho = 0$$

Using lemma 42 to compute the value of $x_{(u,w)}$ as a function of ρ:

$$\begin{aligned}
\sum_{(u,w)\in T} \frac{c_{(u,w)}}{r}(\zeta^v_{uw} + \gamma^v_{uw}\frac{\rho}{r}) + \rho = 0 \\
\sum_{(u,w)\in T} \frac{c_{(u,w)}}{r}\zeta^v_{uw} + \sum_{(u,w)\in T} \frac{c_{(u,w)}}{r}\gamma^v_{uw}\frac{\rho}{r} + \rho = 0 \\
(1 + \sum_{(u,w)\in T} \frac{c_{(u,w)}}{r}\frac{\gamma^v_{uw}}{r})\rho = -\sum_{(u,w)\in T} \frac{c_{(u,w)}}{r}\zeta^v_{uw} \\
(r^2 + \sum_{(u,w)\in T} c_{(u,w)}\gamma^v_{uw})\rho = -r^2 \sum_{(u,w)\in T} \frac{c_{(u,w)}}{r}\zeta^v_{uw} \\
\rho = -(\frac{\sum_{(u,w)\in T} c_{(u,w)}\zeta^v_{uw}}{r^2 + \sum_{(u,w)\in T} c_{(u,w)}\gamma^v_{uw}})r \ .
\end{aligned}$$

□

6.2.1 Example

The computation will be performed assuming that r is positive and as big as we want.

We will apply the one-step method to the example in figure 2 on chapter 1, after generating one zero per row and one per column (see figure 2(b)).

For the sake of simplicity, we will denote the workers as $\{1, 2, 3, 4\}$ and the jobs as $\{5, 6, 7, 8\}$.

Let the initial basis be composed of the arcs $\{(1, 5), (2, 6), (3, 7), (4, 8)\}$, as shown in figure 48.

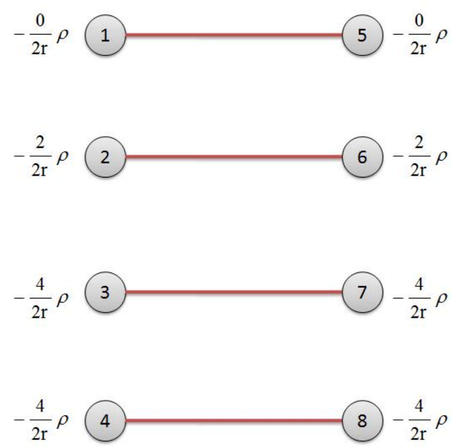

Figure 48: Initial basis for the One-Step NNLS Method.

Calculating the π's as a function of ρ:

$$\begin{aligned}
\pi_1 &= \pi_5 = -\frac{c_{15}}{2r}\rho = -\frac{0}{2}\frac{\rho}{r} \\
\pi_2 &= \pi_6 = -\frac{c_{26}}{2r}\rho = -\frac{2}{2}\frac{\rho}{r} \\
\pi_3 &= \pi_7 = -\frac{c_{37}}{2r}\rho = -\frac{4}{2}\frac{\rho}{r} \\
\pi_4 &= \pi_8 = -\frac{c_{48}}{2r}\rho = -\frac{4}{2}\frac{\rho}{r}
\end{aligned}$$

Calculating the x's as a function of ρ:

$$x_{15} = 1 - \pi_1 = 1 + \frac{0}{2}\frac{\rho}{r} \tag{69}$$

$$x_{26} = 1 - \pi_2 = 1 + \frac{2}{2}\frac{\rho}{r} \tag{70}$$

$$x_{37} = 1 - \pi_3 = 1 + \frac{4}{2}\frac{\rho}{r} \tag{71}$$

$$x_{48} = 1 - \pi_4 = 1 + \frac{4}{2}\frac{\rho}{r} \tag{72}$$

Computing the value of ρ using 69, 70, 71 and 72:

$$\begin{aligned}
\rho &= -\left(\frac{0+2+4+4}{r^2 + 0\frac{0}{2} + 2\frac{2}{2} + 4\frac{4}{2} + 4\frac{4}{2}}\right)r \\
\rho &= -\frac{10}{r^2+18} \times r
\end{aligned}$$

Let us check primal feasibility by substituting the value of ρ in 69, 70, 71 and 72:

$$
\begin{aligned}
x_{15} &= 1 - \frac{0}{2r}\frac{10}{r^2+18} \times r = 1 > 0 \\
x_{26} &= 1 - \frac{2}{2r}\frac{10}{r^2+18} \times r = 1 - \frac{1 \times 10}{r^2+18} > 0 && \text{for } r \text{ big enough.} \\
x_{37} &= 1 - \frac{4}{2r}\frac{10}{r^2+18} \times r = 1 - \frac{2 \times 10}{r^2+18} > 0 && \text{for } r \text{ big enough.} \\
x_{48} &= 1 - \frac{4}{2r}\frac{10}{r^2+18} \times r = 1 - \frac{2 \times 10}{r^2+18} > 0 && \text{for } r \text{ big enough.}
\end{aligned}
$$

Now, let us check dual feasibility:

$$
\begin{array}{rclcl}
\bar{c}_{16} &=& (\frac{0-2}{2}+6)\frac{\rho}{r} = \frac{4}{2}\frac{\rho}{r} & \leq & 0 \\
\bar{c}_{17} &=& (\frac{0-4}{2}+4)\frac{\rho}{r} = \frac{4}{2}\frac{\rho}{r} & \leq & 0 \\
\bar{c}_{18} &=& (\frac{0-4}{2}+2)\frac{\rho}{r} = \frac{0}{2}\frac{\rho}{r} & \leq & 0 \\
\bar{c}_{25} &=& (\frac{-2+0}{2}+5)\frac{\rho}{r} = \frac{3}{2}\frac{\rho}{r} & \leq & 0 \\
\bar{c}_{27} &=& (\frac{-2-4}{2}+3)\frac{\rho}{r} = \frac{0}{2}\frac{\rho}{r} & \leq & 0 \\
\bar{c}_{28} &=& (\frac{-2-4}{2}+0)\frac{\rho}{r} = \frac{-6}{2}\frac{\rho}{r} & > & 0 \\
\bar{c}_{35} &=& (\frac{-4+0}{2}+0)\frac{\rho}{r} = \frac{-4}{2}\frac{\rho}{r} & > & 0 \\
\bar{c}_{36} &=& (\frac{-4-2}{2}+4)\frac{\rho}{r} = \frac{2}{2}\frac{\rho}{r} & \leq & 0 \\
\bar{c}_{38} &=& (\frac{-4-4}{2}+2)\frac{\rho}{r} = \frac{-4}{2}\frac{\rho}{r} & > & 0 \\
\bar{c}_{45} &=& (\frac{-4+0}{2}+4)\frac{\rho}{r} = \frac{0}{2}\frac{\rho}{r} & \leq & 0 \\
\bar{c}_{46} &=& (\frac{-4-4}{2}+0)\frac{\rho}{r} = \frac{-8}{2}\frac{\rho}{r} & > & 0 \\
\bar{c}_{47} &=& (\frac{-4-4}{2}+0)\frac{\rho}{r} = \frac{-8}{2}\frac{\rho}{r} & > & 0
\end{array}
$$

Arcs $(2,8)$, $(3,5)$, $(3,8)$, $(4,6)$ and $(4,7)$ want to enter the basis, as shown by the dotted lines on figure 49. Let us choose the arcs $(4,6)$ and $(2,8)$ to enter the basis. Since they will form a cycle (see figure 51), we must remove arcs $(2,6)$ and $(4,8)$ from the basis (see figure 52). Observe that arc $(3,5)$ can also enter the basis at the same time since it belongs to a different connected component (see figure 50).

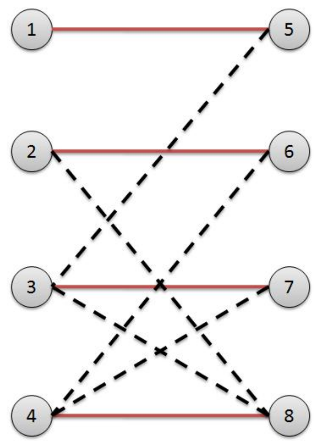

Figure 49: Arcs that want to enter the basis on the first iteration.

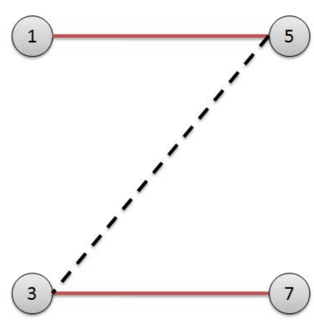

Figure 50: Arc $(3, 5)$ enters the basis.

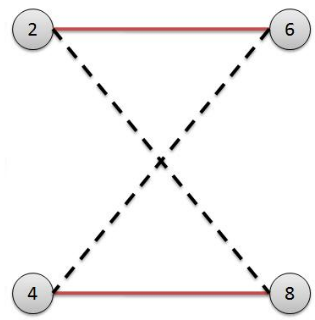

Figure 51: Arcs $(4, 6)$ and $(2, 8)$ enter the basis generating a cycle.

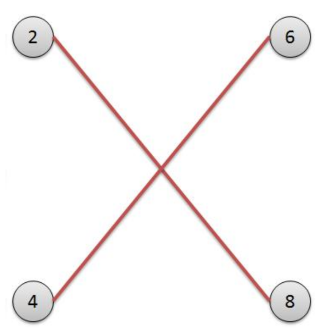

Figure 52: Arcs $(2, 6)$ and $(4, 8)$ are dropped from the basis.

The new basis is shown in figure 53. We now have three connected components: $\{(1,5),(3,5),(3,7)\}$, $\{(4,6)\}$ and $\{(2,8)\}$. Let choose the vertices 1, 4 and 2 as their roots, respectively.

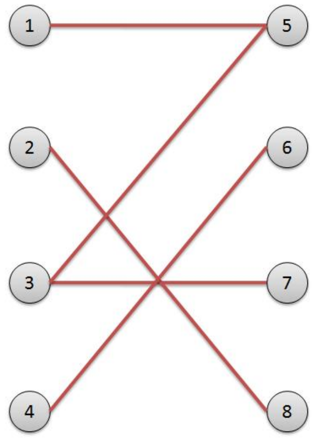

Figure 53: Basis at the end of the first iteration.

For the first connected component, let us compute the π's as a function of π_1 and ρ:

$$\begin{aligned}
\pi_5 &= -\pi_1 + \frac{0}{r}\rho = -\pi_1 + 0\frac{\rho}{r} && (73)\\
\pi_3 &= \pi_1 + \frac{0-0}{r}\rho = \pi_1 + 0\frac{\rho}{r} && (74)\\
\pi_7 &= -\pi_1 + \frac{-2+0-0}{r}\rho = -\pi_1 + (-2)\frac{\rho}{r} && (75)
\end{aligned}$$

Calculating the π_1 as a function of ρ using 73, 80 and 75:

$$\pi_1 = \frac{-2+0+0}{4}\frac{\rho}{r} = \frac{-2}{4}\frac{\rho}{r} \tag{76}$$

Now, substituting the value of π_1 (obtained in 76) in 73, 80 and 75:

$$\begin{aligned}
\pi_5 &= -\frac{-2}{4}\frac{\rho}{r} + 0\frac{\rho}{r} = \frac{2}{4}\frac{\rho}{r} && (77)\\
\pi_3 &= \frac{-2}{4}\frac{\rho}{r} + 0\frac{\rho}{r} = \frac{-2}{4}\frac{\rho}{r} && (78)\\
\pi_7 &= -\frac{-2}{4}\frac{\rho}{r} + (-2)\frac{\rho}{r} = \frac{0}{4}\frac{\rho}{r} && (79)
\end{aligned}$$

Calculating the π's as a function of ρ for the other two connected components:

$$\pi_2 = \pi_8 = -\frac{c_{28}}{2r}\rho = -\frac{0}{r}\frac{\rho}{r} \tag{80}$$

$$\pi_4 = \pi_6 = -\frac{c_{46}}{2r}\rho = -\frac{0}{r}\frac{\rho}{r} \tag{81}$$

Calculating the x's of the first connected component as a function of ρ:

$$x_{15} = 1 - \pi_1 = 1 + \frac{2}{4}\frac{\rho}{r} \tag{82}$$

$$x_{35} = 1 - x_{15} - \pi_5 = 1 - (1 + \frac{2}{4}\frac{\rho}{r}) - (\frac{2}{4}\frac{\rho}{r}) = \frac{-4}{4}\frac{\rho}{r} \tag{83}$$

$$x_{37} = 1 - x_{35} - \pi_3 = 1 - (\frac{-4}{4}\frac{\rho}{r}) - (\frac{-2}{4}\frac{\rho}{r}) = 1 + \frac{6}{4}\frac{\rho}{r} \tag{84}$$

Calculating the x's of the other connected components as a function of ρ:

$$x_{26} = 1 - \pi_2 = 1 + \frac{0}{2}\frac{\rho}{r} \tag{85}$$

$$x_{48} = 1 - \pi_4 = 1 + \frac{0}{2}\frac{\rho}{r} \tag{86}$$

Computing the value of ρ, using 82, 83, 84, 85 and 86:

$$\begin{aligned}
\rho &= -\left(\frac{0+4+0+0}{r^2 + 0\frac{0}{4} + 0\frac{-4}{4} + 0\frac{0}{4} + 0\frac{0}{4}}\right)r \\
\rho &= -\frac{4}{r^2+0} \times r
\end{aligned}$$

Let us check primal feasibility by substituting the value of ρ in 82, 83 and 84:

$$\begin{aligned}
x_{15} &= 1 - \frac{2}{4r}\frac{4}{r} = 1 - \frac{2 \times 4}{4r^2} > 0 \quad \text{for } r \text{ big enough.} \\
x_{35} &= -\frac{-4}{4r}\frac{4}{r} = \frac{4 \times 4}{4r^2} > 0 \quad \forall r \\
x_{37} &= 1 - \frac{6}{4r}\frac{4}{r} = 1 - \frac{6 \times 4}{4r^2} > 0 \quad \text{for } r \text{ big enough.} \\
x_{28} &= 1 - 0\rho = 1 > 0 \\
x_{46} &= 1 - 0\rho = 1 > 0
\end{aligned}$$

Now, let us check dual feasibility:

$$\begin{array}{rclcl}
\overline{c}_{16} & = & (\frac{-2+0}{2}+6)\frac{\rho}{r} = \frac{10}{2}\frac{\rho}{r} & \leq & 0 \\
\overline{c}_{17} & = & (\frac{-2+0}{2}+4)\frac{\rho}{r} = \frac{6}{2}\frac{\rho}{r} & \leq & 0 \\
\overline{c}_{18} & = & (\frac{-2+0}{2}+2)\frac{\rho}{r} = \frac{2}{2}\frac{\rho}{r} & \leq & 0 \\
\overline{c}_{25} & = & (\frac{0+2}{2}+5)\frac{\rho}{r} = \frac{12}{2}\frac{\rho}{r} & \leq & 0 \\
\overline{c}_{27} & = & (\frac{0+0}{2}+3)\frac{\rho}{r} = \frac{6}{2}\frac{\rho}{r} & \leq & 0 \\
\overline{c}_{28} & = & (\frac{0+0}{2}+0)\frac{\rho}{r} = \frac{0}{2}\frac{\rho}{r} & \leq & 0 \\
\overline{c}_{35} & = & (\frac{-2+2}{2}+0)\frac{\rho}{r} = \frac{0}{2}\frac{\rho}{r} & \leq & 0 \\
\overline{c}_{36} & = & (\frac{-2+0}{2}+4)\frac{\rho}{r} = \frac{6}{2}\frac{\rho}{r} & \leq & 0 \\
\overline{c}_{38} & = & (\frac{-2+0}{2}+2)\frac{\rho}{r} = \frac{2}{2}\frac{\rho}{r} & \leq & 0 \\
\overline{c}_{45} & = & (\frac{0+2}{2}+4)\frac{\rho}{r} = \frac{10}{2}\frac{\rho}{r} & \leq & 0 \\
\overline{c}_{46} & = & (\frac{0+0}{2}+0)\frac{\rho}{r} = \frac{0}{2}\frac{\rho}{r} & \leq & 0 \\
\overline{c}_{47} & = & (\frac{0+0}{2}+0)\frac{\rho}{r} = \frac{0}{2}\frac{\rho}{r} & \leq & 0
\end{array}$$

Therefore, this basis is OPTIMAL!

Since the optimal basis will be composed of the arcs $\{(1,5),(3,5),(3,7),(2,8),(4,6)\}$ (as shown in figure 54)), we can extract the perfect match as shown by the thicker lines in figure 55.

Therefore, the solution of the assignment problem will be the perfect matching composed of the arcs $\{(1,5),(3,7),(2,8),(4,6)\}$.

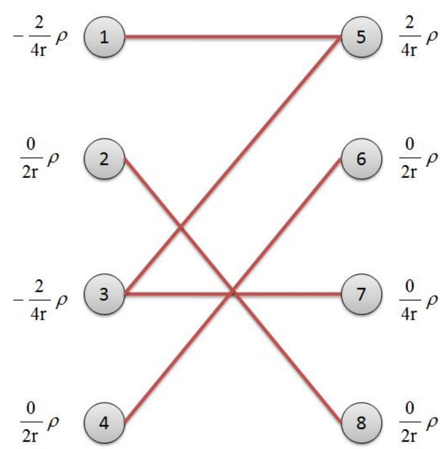

Figure 54: Optimal basis for the One-Step NNLS Method.

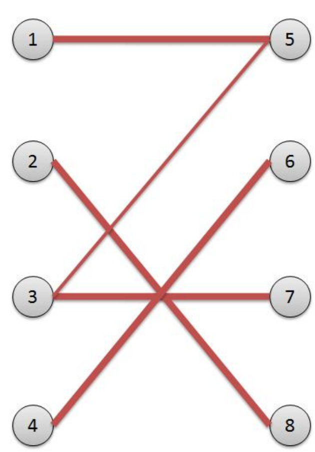

Figure 55: Complete matching of the optimal basis.

REFERENCES

[1] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B., *Network Flows.*

[2] Akgul, M., "A genuinely polynomial primal simplex algorithm for the linear assignment problem," *Discrete Applied Mathematics*, vol. 45, pp. 93–115, 1993.

[3] Balinski, M. L., "Signature methods for the assignment problem," *Operations Research*, vol. 33, pp. 527–536, 1985.

[4] Balinski, M. L., "A competitive (dual) simplex method for the assignment problem," *Mathematical Programming*, vol. 34, pp. 125–141, 1986.

[5] Barnes, E., *Computational Methods - class notes.*

[6] Barnes, E., Chen, V., Gopalakrishnan, B., and Johnson, E. L., "A least-squares primal-dual algorithm for solving linear programming problems," *Operations Research Letters*, vol. 30, 2002.

[7] Bertsekas, D. P., "A new algorithm for the assignment problem," *Mathematical Programming*, vol. 21, pp. 152–171, 1981.

[8] Bertsekas, D. P., "The auction algorithm: a distributed relaxation method for the assignment problem," *Annals of Operations Research*, vol. 14, pp. 105–123, 1987.

[9] Bertsimas, D. and Tsitsiklis, J. N., *Introduction to Linear Optimization.*

[10] Cook, W. J., Cunningham, W. H., Pulleyblank, W. R., and Schrijver, A., *Combinatorial Optimization.*

[11] Dantzig, G. B. and Fulkerson, D. R., "A primal-dual algorithm for linear programs," *Linear Inequalities and Related Systems*, 1956.

[12] Edmonds, J., "Maximum matching and a polyhedron with 0,1-vertices," *Journal of Research National Bureau of Standards Section B*, vol. 69, 1965.

[13] Egerváry, J., "Matrixok kombinatorius tulajdonságairól," *Matematikai és Fizikao Lapok*, vol. 38, 1931.

[14] Goldfarb, D., "Efficient dual simplex algorithms for the assignment problem," *Mathematical Programming*, vol. 33, pp. 187–203, 1985.

[15] H. N. Gabow, R. E. T., "An $o^{2.5}$ algorithm for finding maximum matching in a general graph," *PhD Thesis, Weizman Institute of Science*, 1976.

[16] H. N. Gabow, R. E. T., "Faster scaling algorithms for general matching problems," *Journal of the Association for Computer Machinery*, vol. 38, pp. 815–853, 1991.

[17] Hung, M. S., "A polynomial simplex method for the assignment problem," *Operations Research*, vol. 31, pp. 595–600, 1983.

[18] Hung, M. S. and Rom, W. O., "Solving the assignment problem by relaxation," *Operations Research*, vol. 28, pp. 969–982, 1980.

[19] Kong, S., *Linear Programming Algorithm Using Least-Squares Method*. PhD dissertation, Georgia Institute of Technology, School of Industrial and Systems Engineering, May 2007.

[20] Kuhn, H. W., "On combinatorial properties of matrices," *Logistics Papers, George Washington University*, vol. 11, pp. 1–11, 1955. English translation of Egerváry's paper.

[21] Kuhn, H. W., "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 3, pp. 253–258, 1956.

[22] L. R. Ford, J. and Fulkerson, D. R., *Flows in Networks*.

[23] Leichner, S. A., Dantzig, G. B., and Davis, J. W., "A strictly improving linear programming phase i algorithm," *Annals of Operations Research*, vol. 47, pp. 409–430, 1993.

[24] M. Akgul, O. E., "A dual feasible forest algorithm for the linear assignment problem," *Recherche Opérationnelle*, vol. 25, pp. 403–411, 1991.

[25] Micali, S. and Vazirani, V. V., "An $o(\sqrt{V}e)$ algorithm for finding maximum matching in general graphs,"

[26] Munkres, J., "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, pp. 32–38, 1957.

[27] Orlin, J. B., "On the simplex method algorithm for networks and generalized networks," *Mathematical Programming Study*, vol. 24, pp. 166–178, 1985.

[28] P. Kleinschimidt, C. W. Lee, H. S., "Transportation problems which can be solved by the use of hirsh-paths for the dual problems," *Mathematical Programming*, vol. 37, pp. 153–168, 1987.

[29] Papadimitriou, C. H. and Steiglitz, K., *Combinatorial Optimization: Algorithms and Complexity*.

[30] Paparrizos, K., "A non-dual signature method for the assignment and a generalization of the dual simplex method for the transportation problem," *Operations Research*, vol. 22, pp. 269–289, 1988.

[31] Paparrizos, K., "An infeasible (exterior point) simplex algorithm for assignment problems," *Mathematical Programming*, vol. 51, pp. 45–54, 1991.

[32] R. S. Barr, F. Glover, D. K., "The alternating basis algorithm for assignment problems," *Mathematical Programming*, vol. 13, pp. 1–13, 1977.

[33] Schrijver, A., *Combinatorial Optimization - Polyhedra and Efficiency.*

[34] Vazirani, V. V., "A theory of alternating paths and blossoms for proving correctness of the $o(\sqrt{V}e)$ general graph maximum matching algorithm," *Combinatorica*, vol. 14, 1994.

[35] Wolsey, L. A., *Integer Programming.*